

**AN EFFICIENT AND SECURE MECHANISM FOR  
DATA SHARING FROM MOBILE PHONES TO SERVER**



**MS Research Dissertation**

**Submitted By**

**Rukhsana Kousar**

**(438-FBAS/MSCS/S08)**

**Supervised By**

**Prof. Dr Muhammad Sher**

**Co-Supervised By**

**Dr. Zeeshan Shafi Khan**

**Department of Computer Science and Software Engineering**

**Faculty of Basic and Applied Sciences,**

**International Islamic University, Islamabad**



TH-8875

Accession No.           

MS

005.73

RUE

1. Data stores

2. Abstract data types

DATA ENTERED

Amz 10/04/13

A Dissertation submitted to the  
**Department of Computer Science**  
**and Software Engineering**

International Islamic University Islamabad

As a partial fulfillment of requirements for the award of

The degree of

**MS in Computer Science**

**INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD**  
**FACULTY OF BASIC AND APPLIED SCIENCES**  
**DEPARTMENT OF COMPUTER SCIENCE & SOFTWARE ENGINEERING**

Dated: 17-01-2012

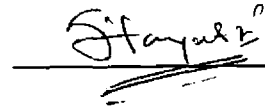
**Final Approval**

It is certified that we have read the thesis titled “An Efficient and Secure Mechanism for Data Sharing from Mobile Phones to Server” submitted by Ms. Rukhsana Kousar Registration No.438-FBAS/MSCS/S08. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the degree MS in Computer Science.

**COMMITTEE**

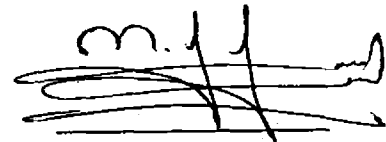
**External Examiner:**

**Dr. Sikandar Hayat Khiyal,**  
Professor/Head of Acad (ES),  
APCOMS,  
Khadim Hussain Road,  
LalKurti Rawalpindi



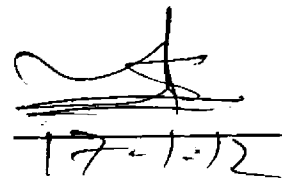
**Internal Examiner:**

**Mr. Muhammad Imran Saeed,**  
Assistant Professor,  
DCS&SE, FBAS, IIUI



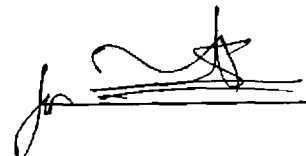
**Supervisor:**

**Prof. Dr. Muhammad Sher,**  
Chairman,  
DCS&SE, FBAS, IIUI



**Co-Supervisor:**

**Dr. Zeeshan Shafi Khan,**  
Senior Researcher,  
King Fahad University Saudi Arabia



## **Declaration**

We hereby declare that this work, neither as a whole nor as a part has been copied out from any source. It is further declared that we have conducted this research and have accomplished this thesis entirely on the basis of our personal efforts and under the sincere guidance of our supervisor Prof. Dr Muhammad Sher and our Co-Supervisor Dr. Zeeshan Shafi Khan. If any part of this project is proved to be copied out from any source or found to be reproduction of some other project, we shall stand by the consequences. No portion of the work presented in this dissertation has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

**Rukhsana Kousar**

**(438-FBAS/MSCS/S08)**

## **Dedication**

**Mr. Tariq Mehmood:** Bundle of thanks for support and encouragement as well as advices that enabled me to manage my aspiration. You are precious bequest of ALLAH (S.W.T).

**My Parents:** Bundle of thanks for supporting me, affection and steady love and believing in me. I will try my best to attain all your dreams.

**My Teachers:** Bundle of thanks for commitment, support and guidance that make possible my study.

**My Siblings:** Bundle of thanks for cheering me, soothing me, always is with me, and raising my trust in Allah ﷻ (S.W.T) better knows what is best for us.

## **Acknowledgement**

In the name of Allah who is more gracious and more merciful.

All extol and appreciation to Allah ﷻ (S.W.T) most merciful most compassionate, for Allah ﷻ (S.W.T) has power over everything. We exalt, ask for his help, pity and his fortification in opposition to our evil-self also depraved doing. I want owing my profound sense of gratitude to Allah ﷻ (S.W.T) who make possible to complete this thesis by countless favor and bounty. The attempt finished by belief in Allah ﷻ (S.W.T) and faith on Holy Prophet Muhammad ﷺ (PBUH) forever turn out well.

“Mentioning (Speaking of) the favors of Allah ﷻ (S.W.T) is (a show of) gratefulness. Leaving it (the favor) is ingratitude. Whoever does not thank (for) the little will not thank the much. And he who does not thank the people does not thank Allah ﷻ (S.W.T).....” (On the authority of Nu'man Ibn Basheer, Hadith no 5325 in AlJami' Assaghir). In the production of this dissertation many people have contribute while only my name is appear on the cover of this dissertation. I would like to pay my gratitude to all those people who helped me to complete this dissertation.

My acknowledgment as well goes to my supervisor Prof. Dr Muhammad Sher with deepest feelings of great gratitude on behalf of his comprehensible supervision and moral support throughout the study and made my research work easier and achievable.

I am deeply grateful to my co-supervisor Dr. Zeeshan Shafi Khan whose support and encouragement as well as advices enabled me to develop an understanding of this

subject. I am heartily thankful to him for the reason that without his collaboration and support it will be difficult for me to attain my goal.

I am furthermore gratified to the following former and present staff at IIU, for their support for the duration of my study. I am also grateful to the following former or current staff at IIUI, for their various forms of support during my Studies and especially during this research.

Most significantly, love and patience of my family make this possible; especially my mother has been a constant source of love, concern support and strength all these years. I would like to state my heartfelt gratitude to my family. I deeply appreciate Mr. Tariq Mehmood who helped me stay sane during these difficult years. I would like to admit that His support and concern helped me to complete this work and help out to overcome setback and keep on focused on my study. He and my siblings and especially my mother were always here cheering me up and stood by me all the way through the good times and bad.

I am profoundly grateful to my sincere friends for their involvement and support in my study. Finally I state my inestimable gratitude to all those people who have helped me during completion this MS degree because without their help and guidance I would never have been able to finish my thesis.



My utmost and eventual gratitude is due to Allah ﷻ (S.W.T), the creator of heavens and earth. May he forgive my flaw and weakness, make stronger and enliven my belief in him and bestow me with knowledge, awareness and wisdom, ameen!

Rukhsana Kousar

## **Project in Brief**

<b>Project Title:</b>	<b>An Efficient And Secure Mechanism For Data Sharing From Mobile Phones to Server</b>
<b>Undertaken By:</b>	<b>Rukhsana Kousar</b>
<b>Supervised By:</b>	<b>Prof. Dr Muhammad Sher</b>
<b>Co-Supervised By:</b>	<b>Dr. Zeeshan Shafi Khan</b>
<b>Start Date:</b>	<b>February 2009</b>
<b>Completion Date:</b>	<b>January 2012</b>
<b>Tools and technologies:</b>	<b>MS Visual Studio 2008 C# .net and MS SQL Server 2008</b>
<b>Documentation Tools:</b>	<b>MS Word, EDraw, MS Excel, MS Visio</b>
<b>Operating System:</b>	<b>MS Windows XP professional</b>
<b>System used:</b>	<b>Pentium 4, Intel core I 5</b>

## **Abbreviations Used**

<b>Abbreviations</b>	<b>Acronyms</b>
ICMP	Internet Control Message Protocol
TCP	Transmission Control Protocol
PKI	Public Key Infrastructure
IMS	IP Multimedia Subsystems
SHA	Secure Hash Algorithm
DHT	Data Hash Table
NFS	Network files System
AFS	Andrew files system
NASD	Network attached storage system
CFS	Cryptographic files system
MAC	Message Authentication Code
DES	Data Encryption Standard
AES	Advanced Encryption Standard
PHS	Personal Handy-Phone System
P2P	Peer-to-Peer
GPRS	General Packet Radio Service
WAP	Wireless Application Protocol
DDoS	Distributed Denial-of-Service
EFS	Encrypting File system for window
SFS	Self-certifying File System

## **Abstract**

Mobile phones are battery operated as well as have limited storage capacity. So due to limited storage it is preferred to store some portion of the contacts on storage server and retrieve them whenever required. Storing contacts in plain text invites the attacker to hack or attack the data. So it is stored in cipher text. Because mobile phones are battery operated so it is preferred to encrypt the contacts on server to save the battery of the mobile phones. Shifting and encrypting contacts on server raises various security concern including data sniffing, modification denial of service etc. To solve all these issues we proposed a mechanism that will securely transfer the data from mobile phones to server and secures the server from denial of service attack.

# **Table of Contents**

<b>#</b>	<b>Contents</b>	<b>Page #</b>
<b>1.</b>	<b>Introduction</b>	<b>02</b>
1.1	Symmetric Key Cryptography	04
1.1.1	Symmetric Cipher Model	05
1.1.2	Symmetric key algorithm	06
1.1.3	Cryptographic Primitives Based on Symmetric Cipher	06
1.1.4	Security of Symmetric cipher	07
1.1.5	Key Generation	07
1.2	Asymmetric Key Cryptography	08
1.2.1	Public Key Cryptosystem	08
1.2.2	Application for the Public key cryptosystem	10
1.2.3	Symmetric vs. asymmetric algorithm	11
1.3	Block Cipher	12
1.4	Stream cipher	12
1.4.1	Stream cipher vs. Block cipher	13
1.5	Hash Function	14
1.5.1	SHA Hash Function	15
1.5.2	SHA-0 and SHA-1	15
1.5.3	SHA-2 Family	16
1.5.4	SHA-2 is Better from SHA-1	17
1.5.5	SHA-3	17
1.6	Distributed Denial of Service Attacks	17

1.6.1	DDoS Attack Depiction	18
1.6.2	Constructing the Attack Network	21
1.6.3	DDoS attack Countermeasures	22
1.7	Thesis Outline	23
2.	Literature Survey	25
2.1	Limitation of the Literature Surveyed	35
2.2	Summary	37
3.	Requirements Analysis	39
3.1	Problem Definition	39
3.2	Research Objectives	42
3.3.1	Utilization of Resource	42
3.3.2	No Denial of Service	43
3.3.3	Cost Effective and Enhance Applicability	43
3.3.4	Minimize the Chances of Flooding Attack	43
3.3.5	Minimize the Loss	43
3.3.6	Provide an efficient service	43
3.3.7	Increasing the overall Data sharing security	43
3.3.8	Evaluation and Formulation	44
3.4	Summary	44

<b>4.</b>	<b>Proposed Solution and Methodology</b>	<b>45</b>
4.1	Authentication and Data Sharing	49
4.2	Calculation of Amount of Data Transferred	53
4.3	Securing the Data in Transit from Mobile Phone to Server	
4.4	Summary	61
<b>5.</b>	<b>Results</b>	<b>62</b>
5.1	Scenario 1: Data Transfer Rate vs. Battery Lives (Sufficient Storage)	63
5.2	Scenario 2: Data Transfer Rate vs. Battery Lives (Less Storage)	65
5.3	Summary	71
<b>6.</b>	<b>Conclusions and Future Work</b>	<b>72</b>
6.1	Conclusion	72
6.2	Future Work	73

# Chapter 1

## Introduction



## 1. Introduction

A mobile phone is an electronic device is used for mobile voice video and data. Now a day's wireless communication becomes a part of our daily life, however privacy and security issues are arising. Mobile privacy and security becomes a core issue in society, have a number of factors. In human society mobile phones play a vital role. People are relying on mobile communication. Exchange of data through mobile phones, we can say that it provides a new exchange platform. But during the transformation of data, the information is open to the elements of threat. Individual privacy and personal data information security has increased the concern by business and personal users. Mobile phone security and privacy issue common and major concern when transactions are made by internet. Unauthorized access or eavesdropping and stealing data or information are main security threats in the existing mobile phones network presently. Two components involve in this problem: identification integrity and message integrity. No modification an altering the contents or attempt to open the message by a third party just receiving the message as sent refer the message integrity. Message is originating with the signature element in order to establish the identification integrity. Possible technical countermeasures: for registration or roam network subscription GSM network is authenticated. At one particular time only one user can make the expected response subscriber identity card (SIM) and home location register (HLR) have some shared secretes. Biometrics checking is one possible countermeasure. For encryption the existing algorithm can be broken down by hackers so secure the information completely is impossible or there is no tool which provides a complete security. Few decades ago it is impossible or impractical to provide the services when and where they want and deliver a personalized location sensitive content and application to user by mobile devices.

At the present time much more personal device is mobile phone and personal computers become less significant as compare to it. PC did only computation and application while mobile phone provide an internet services and flexible access to communication at home, office, car and easily can travel with users. 2G network provide a limited ability against security threat. Major security threats come out in mobile devices when these devices use the internet. GPRS (General Packet Radio Service) and WAP (wireless application protocol) provide the internet accessing service to mobile communication carrier in the GPS environment to protect the transformation of information and data. As compared to GSM network the greater data transmission ability provide by the 3G mobile communication system. Enhanced protocol and due to a greater bandwidth the next generation mobile communication system will turn out to be supposed mobile internet. IPv6 protocol and Wi-Fi a technology on mobile phones is affected by virus and also increases the security threats. Mobile terminals like PHS(Personal Handy-phone System), cellular phone, notebook PCs etc are simply carried out and they are forever on that's why users of mobile terminals boost up day by day, these terminals are extensively used for storing the business information . Data is stores on these terminals in different format but when they shares a common data among PC and cellular phone and PHS then face a trouble due to different data format. Gmail, hotmail, YouTube etc are some methods to solve such problems, these methods present storage services provided by the network services which are commonly used but this type of storage services is not preferable or suitable for the treatment of business information. Applications of the new generation consider the distributed storage mechanism as de facto method of data storage. Performance, availability, scalability and other several motive of distributed storage system that suppress the traditional database system. Distributed processing is the power of web companies for example Google Amazon and yahoo. Terabytes and even pentabytes of data processing are required for a new generation

application. Multi core CPU architecture as well as multiple systems gives the chance /opportunity to program that should be scalable and take advantages from them. Distributed data implies by distributed processing. An architectural era is almost end by the suggestion of several researchers and vendors have to start over relational database system. In a business application RDBMS still have their place but next generation applications change the requirements of storage system for web services. Distributed storage system have some common principles save, see, secure, share are four ingredients of data management. Loosely structured data is available there is no rigorous schemes. Each server achieved the speed and latency by ensuring the optimal size of data and reducing the number of network hops.

In 2008 distributed storage system is anticipated for defensive confidential information stored in mobile handsets. This storage system distributes the information into two parts: one part is stored at the network storage (home server) and other at the storage of mobile terminals by using the self encryption scheme. Without any users intercession in self-encryption scheme exclusive/unique keys are automatically generated and after being used it is automatically deleted. Concerning the key management users don't worry about it because terminal use key only for internal processing. Mobile terminals have limited resources but a user makes effective use by selecting the file splitting algorithm. Communication band and the character of target file and also size of the user file help to choose the encryption and key generation according to the situation.

## 1.1 Symmetric Key Cryptography

Symmetric key cryptography [19, 20] is a conventional encryption scheme in which sender and recipient shared a solitary one pair of strategy key. Only recipient knows the symmetric key cryptography in the asymmetric key cryptography infrastructure. Symmetric key

cryptography is a conventional encryption or enciphering scheme, some terms we should define: the first one is plaintext that is original intelligible data or message that is fed into algorithm as an input; scrambled message or coded message is known as cipher text. Encryption is a process of converting plain text into cipher text and deciphering or decryption is a process of converting cipher text into a plain text. There are many schemes used for encryption constitute the study area recognized as cryptographic system. Cryptanalysis is the area in which without any knowledge of enciphering details some technique used for deciphering the message.

### 1.1.1 Symmetric Cipher Model

A symmetric cipher model has five ingredients for symmetric encryption scheme:

**Plain Text:** Plaintext is the original intelligible message or data that is treated as input for the algorithm.

**Encryption Algorithm:** Various substitution and transformation performed by the encryption algorithm on plain text.

**Secret Key:** Secret key value is independent of the plain text and algorithm. Encryption algorithm also take secret key value as input. The algorithm depends on the key for performing the exact substitution and transformation. So the different output of the algorithm depend the specific secret key used at that time.

**Cipher Text:** Cipher text is a coded or scrambled message produced by the encryption algorithm as an output. Two different cipher text produced by two different keys for a given message. So cipher text output is dependent on secret key and plain text. Apparently cipher text is a random data streams and is unintelligible as stands.

**Decryption Algorithm:** Encryption algorithm run in reverse it produces the original plain text by taking the cipher text and specific secret key.

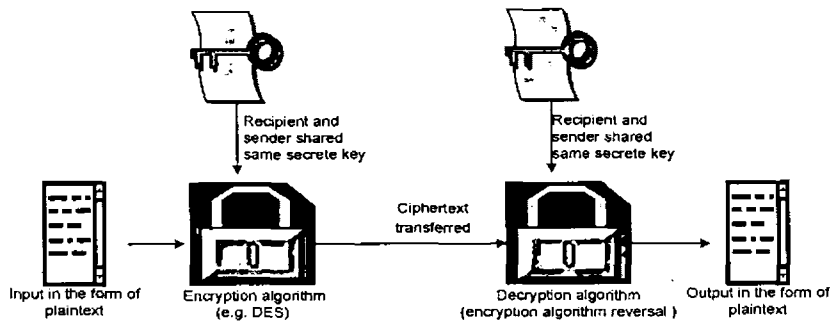


Figure 1.1: Symmetric Cipher Model [19]

## 1.1.2 Symmetric Key Algorithm

Algorithms have a class of symmetric key algorithm for cryptography that use slightly related. Often identical encryption and decryption both use cryptographic keys. Secret key, single key, shared key, one key and private key encryption are other terms used for symmetric key encryption. For maintain a private information link between two or more parties used a shared secret key. Encryption and decryption key may be identical or trivially related to each other or between these two keys simple transformation to go down. Block cipher or stream cipher are two categories of symmetric key algorithm. encrypt the message bit by bit or byte by byte at a time by using the stream cipher, while number of bytes taken as a input by a block cipher it treat them as a single unit for encryption. 64-bit block have been commonly used, 128-bit block used by advanced encryption algorithm approved by NIST in December 2001. Twofish, serpent, AES (Rijndael), Blowfish, CAST5, RC4, TDES and IDEA are some illustrious examples of symmetric algorithm.

## 1.1.3 Cryptographic Primitives Based on Symmetric Cipher

Symmetric cipher used for encryption but often used for achieving other cryptographic primitives. At the receiver side changes can be noted down by using the MAC (Message

authentication code) that is added with cipher text because message can be changed while it is encrypted, message encryption does not provide a guarantee that this message is not altered. Symmetric cipher is used for the construction of message authentication code. e.g. [CBC-MAC], Non-repudiation also used to build Hash Function. Many modern block ciphers are based on construction proposed by Horst Feistle. Feistel's construction allows building invertible function that are they not invertible.

#### 1.1.4 Security of Symmetric Cipher

Chosen plaintext, differential cryptanalysis, linear cryptanalysis plaintexts are historically susceptible by symmetric cipher. Chances of successful attack can greatly reduced by the careful construction of function of each round.

#### 1.1.5 Key Generation

For the symmetric cipher session key, pseudorandom key generator is used. Though, lack of randomness in these generators has led to cryptanalytic breaks in the past. Hence it is necessary initializing resources should be high entropy used for implementation. For encrypting information there are two basic techniques that are used. Symmetric encryption (also called private key or secret key encryption) and asymmetric encryption (also called public key encryption)The best one and oldest technique is the symmetric encryption, a word, number or just a string of random letters can be a secret key that is applied to text of the message that change the content in a particular way. This may be as easy as changing every letters places in the alphabet. Symmetric key known by both sender and recipient and they can encrypt and decrypt all messages by using this key.

## 1.2 Asymmetric Key Cryptography

Asymmetric key cryptography [19, 20] is nonreversible encryption algorithm. Two kinds of keys used in this scheme: one is asymmetric key which is well-known to all and only the receiver knows another key that is symmetric key or secret key. When secrete key is exchanging over the internet or a large network face a problem while preventing them from failing into wrong hands. The message can be decrypted by anyone who knows the secrete key. Asymmetric encryption is the answer of this problem. In this scheme a pair of key or two related keys are used. Private Key only you know it that's why kept secret and another key is public key is made freely available to anyone who want to send you a message. By using public key any message (text, binary files and documents) can be encrypted and through using the matching private key applying the same algorithm for decrypting the message. By using the private key if message is encrypted then this is only decrypted by using the matching public key.

### 1.2.1 Public Key Cryptosystem

A symmetric algorithm use two keys one for encryption and other for decryption.

A public key encryption scheme has six components.

**Plain Text:** This is input (graspable message) that gives to the algorithm for processing.

**Encryption Algorithm:** Many transformations perform on this input by encryption algorithm.

**Private and Public Key:** Public key is used for encrypt the message and private key is used for decrypt that message, both keys known as pair of keys. Algorithm performs how much exact transformation on message depends on provided public or private key.

**Cipher Text:** This is an output in the form of scrambled message that depends on one type of key and plain text.

**Decryption Algorithm:** This algorithm is used for attaining plaintext from cipher text. For producing the original plaintext it necessitate matching key and cipher text.

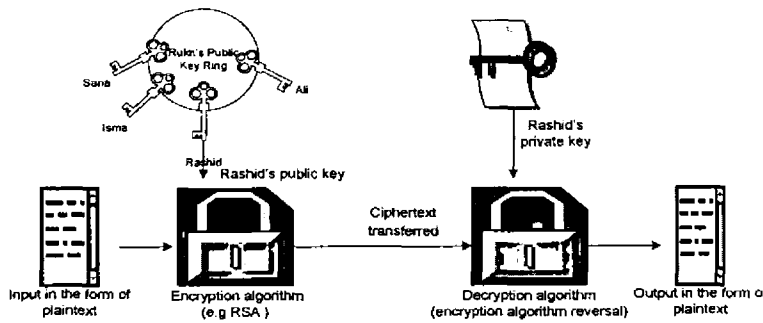


Figure 1.2: Encryption of Public Key [19]

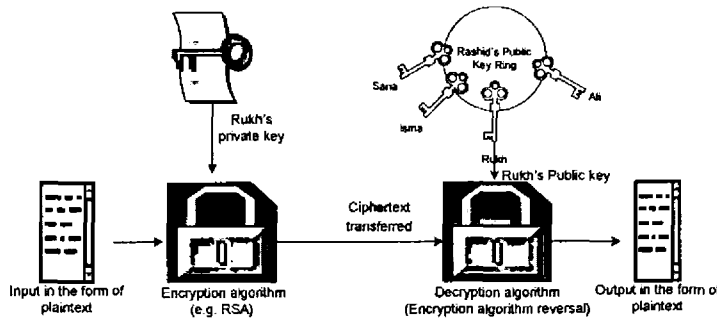


Figure 1.3: Public Key Authentication [19]

The following are the essential steps:

1. For the encryption and decryption of message each user generates a pair of keys.



2. Each user should maintain public keys collections that obtain from others. Every user should place public keys in accessible file or public register and companion key should be kept.
3. If Rukh want to send secrete message to Rashid, Rashid's public key is used for encrypting the message of Rukh.
4. Rashid used his private key to decrypt the message that received from Rukh. This message can't be decrypt by anyone except Rashid.

When we use symmetric key cryptography system it can be a cause of major vulnerability because some secrete key is used for both encryption and decryption while this is not an issue in asymmetric key cryptographic system. Both cryptographic systems have their own strength and weaknesses symmetric system execution is fast as compared to asymmetric cryptographic system. Masquerade attack take place in asymmetric due to vulnerability. Yet they are important and have fussy benefits. Both system can work together and create a cryptographic mechanism that provide high level security and work efficiently.

### 1.2.2 Application for Public Key Cryptosystem

Cryptographic algorithm of public key system use two keys one is private and other is public.

Cryptosystem of public key classify into three categories.

**Encryption /Decryption:** Recipient public key is used to encrypt the sender message.

**Digital Signature:** Private Key of sender is used to "sign" the message. A cryptographic algorithm applied on message for achieving the signing.

**Key Exchange:** Session keys to exchange two sides cooperate.

Examples of asymmetric cipher:

## RSA

For constructing key pairs RSA uses large prime numbers. RSA cryptographic working explained by RSA laboratories as follow: P and q are two large prime numbers, modulus is n and their computed product is  $n=pq$ . e is the number that is less than n and only known by the sender which is relatively prime to  $(p-1)(q-1)$  that means except 1, e and  $(p-1)(q-1)$  have no common factor. Another number is d only receiver knows its value that  $(ed-1)$  is divisible by  $(p-1)(q-1)$ . Value e that is known by sender is also called public exponent and d is only known by receiver is also called private exponent. When anybody knows a public key he can obtain a route of private key that depends on factoring of modulus and its prime components. Choosing keys of adequate length is difficult and made essentially impossible measurement is based on modulus length. For general corporate 1020 bit key size is generally recommended from RSA laboratories and for extremely vulnerable material double key size is recommended. 768 bits key length for ordinary use is adequate and by using current techniques these can't be broken. RSA laboratories mention factoring technique that is a base of RSA key size security.

### 1.2.3 Symmetric vs. Asymmetric Algorithm:

For encryption and decryption the same key is shared by both parties when they use symmetric algorithms. There is a need to be kept secret these keys for providing privacy because it is not safe anymore if anyone else gets to know to the key. Computing power is not consuming too much in these symmetric algorithms is the advantage of this technique. DES, Triple-DES (3DES), IDEA, CAST5, BLOWFISH, TWOFISH are few well known examples. Pair of keys used in asymmetric algorithm one key is used for encryption and other key is used for decryption. Encryption key is called Public key because this key is also spread to all who might want to send encrypted message and decryption key is also called private key

because it typically kept secretly. Everybody who has public key can encrypt the message and send to the owner of secret key. Public key can't help to reconstruct the secret key. Diffie and Hellman give the idea of asymmetric algorithm and it was first published in 1976. For real world use asymmetric algorithms seem to be ideally suited. The risk of getting know is much smaller because secret key does not have to be shared. Every user has a one secret key and collection of public keys. Public keys need only to be protected against being changed while only one key needs to keep secrecy. RSA, DSA, ELGAMAL are some well known asymmetric algorithm. Every pair of users would need to have an own shared secret key with symmetric keys. Symmetric algorithms are much faster as compare to asymmetric algorithms. Therefore, combination of both is being used in many applications. By using asymmetric encryption one or more symmetric keys are generated and exchanged. Asymmetric keys also used for authentication, the advantage of both algorithms can be used by this way. RSA/IDEA combination of PGP2 or the DSA/BLOWFISH used by GnuPG is the example of this typical procedure.

### 1.3 Block Cipher

Block cipher [19] is a symmetric encryption algorithm which transforms the 64 or 128 bits block of plain text into same length of cipher text. DES AES IDEA is the example of block cipher. The mechanism of block cipher consists of two algorithms one is used for encryption that accepts an input of n-bit block and size of key is k-bits, for any one of predetermined key squashes (yielding) n-bits of output block and another for decryption in which the whole process of encryption is converse

$$E_k^{-1}(E_k(M)) = M$$

$E_K$  is permutation for every key beyond the set of input block. Each key choose one permutation as of the possible set of  $2^n!$   $M$  represent block of data,  $E$  used for encryption,  $E^{-1}$  used for decryption.

## 1.4 Stream Cipher

Stream cipher [19] is as well a symmetric encryption algorithms in which stream of plaintext input generate the bit by bit or byte by byte cipher text output. RC4; A5/1 is some examples of stream cipher. Stream cipher composition represent by a diagram. Apparently 8-bit arbitrary number stream (key stream) produces through a pseudorandom bit generator by giving key input. At a time one bit or byte of plain text is combined with that one key stream by using bitwise exclusive XOR operations and resulting cipher text byte is obtained.

```

10101100  plain text
11001100  key stream
-----
01100000  cipher text

```

This is the encryption format of stream ciphers. For decryption mechanism same key stream and pseudorandom sequence is combined with that one cipher text and apply an exclusive OR operation for obtaining a plain text.

Like

```

01100000  cipher text
11001100  key stream
-----
10101100  plaintext

```

### 1.4.1 Stream Cipher vs. Block Cipher

Consistent transformation and fixed digits of large blocks of operation present by block cipher. Block cipher execution is slower than the stream cipher and hardware complexity is high as compare to stream cipher. If stream cipher used incorrectly then face a serious security problem. Same key stream never use a twice, must be used different keys for securely using a synchronous stream cipher. Authenticity does not provide by stream cipher, application designer recognize this point. During transition encrypted message still have been modified, wireless connection etc type of application in which unknowable length of plaintext comes so stream cipher is suitable or used for this type of application E.g. if 32-bit plaintext received by 128-bit block so there would be a three quarters of padding bits of the data transmitted. While stream cipher manage the smallest units of data that can be transmitted (usually bytes) and reduce this concern. In block cipher change of one bit in the plaintext or in the key cause many bit changes in the cipher text e.g. strong avalanche effect exhibit by DES, while in stream cipher change of one bit effect only one bit or one byte of the cipher text remaining data is correct. Mobile handsets use stream cipher except block cipher based on these characteristics which we discussed above.

### 1.5 Hash Function

A function [19] which serves as the authenticator that maps a message of any length into a fixed length hash value. A function  $H$  generated the hash value  $h$

$$h = H(M)$$

The  $H(M)$  is fixed length hash value where  $M$  is a variable length of message. The message is assumed to be correct when the hash value is appended to the message at the source. For

the protection of hash value some means is required because the hash function not be secreted. By re computing the hash value receivers authenticate the message.

### 1.5.1 SHA Hash Function

SHA stand for secure hash Algorithm. SHA hash function designed by National Security Agency and published by the NIST as a U.S federal information processing standard. SHA hash function is a set of cryptographic hash function but prepared differently these are SHA-0, SHA-1, SHA-2. SHA-224, SHA-256, SHA-384 and SHA-512 are the variable digit size of, algorithm used in SHA-2 family. Among the existing hash function best established and employed in numerous widely used security application is SHA-1. Electronic data (message) computes by four secure hash algorithm SHA-1, SHA-256, SHA-384, SHA-512 specified by 180-2 standard. SHA-1 and SHA-256 treat the message of any length  $<2^{64}$  bits input and SHA-384 and SHA-512 treat the message of any length  $<2^{128}$  bits and their output is called message digits. 160-512 bit length is the range of message digit this is depends on algorithm. In software, firmware, hardware, or any arrangement secure hash algorithm can be implemented.

### 1.5.2 SHA-0 and SHA-1

In SHA-1[16] compression function within one iteration 32-bit word of the state are A, B, C, D, E and nonlinear function varies that is F. for each operation n varies, left bit rotation denoted by  $\lll n$  message expanded word denoted by  $w_t$  of round t, constant round represent by  $K_t$  of round t. addition modulo  $2^{32}$  denote by  $\oplus$ .

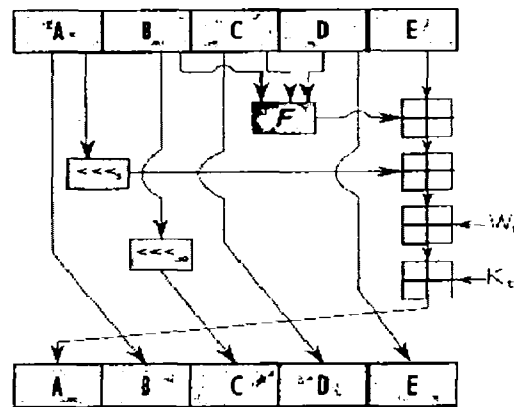


Figure 1.5: SHA-1[16]

By using SHA-1 and SHA-0 message with the maximum length of  $(2^{64}-1)$  bit produces 160-bit digits. FIPS PUB 180 published the original specification of algorithm in 1993 as a secure hash standard by the US government standard agency NIST (National Institute of Standard Technology) now often referred this version as SHA-0; after publication NSA was withdraw this version. The revised version was superseded by SHA-0 and published in FIPS PUB 180-1 in 1995 commonly referred as SHA-1 the message schedule of SHA-1 compression function differ from SHA-0. Cryptography security is reduced in the original algorithm by correct the flaws. Later on SHA-0 and SHA-1 both reported weakness.

### 1.5.3 SHA-2 Family

In 2001 the review and comments were accepted and published in the draft FIPS PUB 180-2. In 2002 FIPS PUB 180-2 released official standard which includes SHA-1 ,In FEB 2004 FIPS PUB 180-2 was published a change notice specifying SHA-224 is additional variant.32-bit and 64-bit words are computed with novel hash function that are SHA-256 and SHA-512. These functions are virtually identical but use a different shift amount, additive constant and different number of rounds.

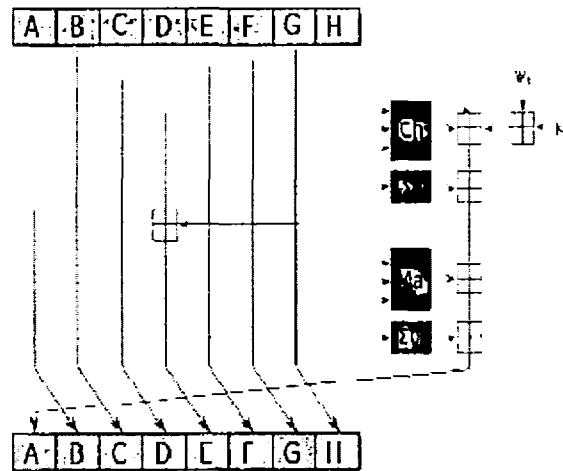


Figure 1.6: SHA-2[17]

The truncated version SHA-384 and SHA-224 of these two functions are computed with two different initial values. SHA-2 family provides a better security but is not widely used as SHA-1 because uses of protocol like SSL in a new hash function don't make it easy without breaking backward compatibility. For the authentication of Debian Linux Software Package and DKIM message signing standard SHA-256 is used. International criminal Tribunal of the Rwandan Genocide authenticates the archival video by the SHA-512 which is used as a part of this system.

#### 1.5.4 SHA-2 is Better from SHA-1

In the sense of security and resources (e.g. if our message size is  $2^{64}$  and we use SHA-1 it provide us 80 bit number of steps while when we use SHA-2 family (SHA-256) for this message length we achieve 128-bit security in 64 number of steps and also output size is 256-bit while SHA1 give 160-bit message digit size /output size) We take input message with maximum length of less than  $2^{128}$  bit. And we use SHA1 then number of blocks increases that



definitely time consuming and if we use SHA2 family in which the standard SHA-512 and its truncated SHA-384 with input processing in 1024 bits blocks and produce output a 512 bit message digest in 80 no of steps. So SHA2 family provides more security and definitely takes less time and resources. Attacks are found in SHA-1 in February 2005 while no attacks yet have been reported on SHA-2 family.

### 1.5.5 SHA-3

On nov2, 2007 NIST hash function competition was formally announced an open competition for new SHA-3[18] function in the Federal register. Through public competition NIST is initiating one or more additional Hash Function to develop like or similar to an advanced encryption standard to the development process. Publication of new standard and proclamation of winner are scheduled to take place in 2012.

## 1.6 Distributed Denial of Service Attacks

Distributed denial of service attacks [19] is growing rapidly. These present a significant security threat to corporations. Legitimate users can't access the resources by useless traffic or flooding server that make computer system inaccessible. This is called distributed denial of service attack, useless packets sends from a large number of compromised hosts that are summative in Distributed denial of service attack. Real attacker recognition is more difficult because attack methods and tools are more effective and sophisticated. Defensive technology is incapable to endure large scale attack. A denial of service attack is an endeavor to prevent intended user from the computer resources. Generally denial of service attack consist effort of user to thwart an internet site or service of target server. Legitimate traffic can't access the resources because external communication request involves saturating the target. DDoS cause

more severe threat. DDoS attack target is attacked in coordinate approach, numeral host or devices collectively launch DDoS attack.

### 1.6.1 DDoS Attack Depiction

In DDoS attack target resources consumed hence it can't provide the services. In the local network internal host consumed the resources of target system. Fig shows the internal resources attack that is SYN flood attack.

1. Targeted server received the connection request over the internet from multiple hosts that are under control of attacker.
2. Target receives TCP/SYN (synchronization/ initialization) packets from the host with erroneous IP address.
3. TCP connections open at each SYN packet that is request. Target server reply and trying to establish TCP connection with SYN/ACK packets with suspicious IP address. Legitimate connection are denied because for each SYN request target server maintain data structure and waiting response back for completion of bogus half open connection's more traffic flood in it becomes bogged down.

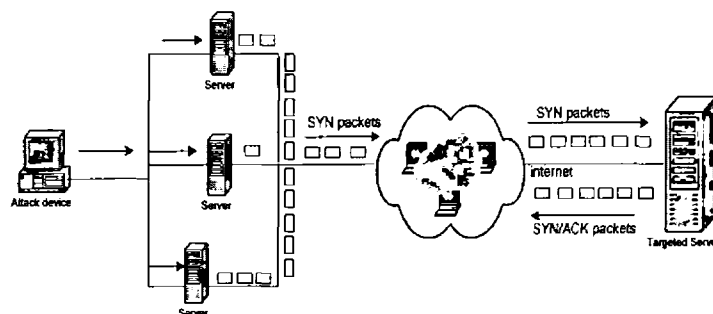


Figure1.7: Flood Attack of SYN in Distributed Environment [19]

The example of internal resources target is TCP data structure.

1. consume a disk space including
  - Anonymous network shared area holding files.
  - Mail message generate in excessive number.
  - Errors intentionally generate.
2. In many systems for processing the information like process table entries, identifiers and slots etc. limited data structure available to hold. These data structure consume by writing a simple data structure or script of intruder that create replication itself and does nothing.

Fig shows data transmission resources consume by an attack.

1. Over the internet multiple hosts under the control of attacker and it instruct them to send ICMP echo packets to a group of host with target spoofed IP address that acts as reflectors.
2. Target site receive the echo reply packets that are response against the multiple spoofed request from the nodes at bounce site.
3. for legitimate traffic no data transmission capacity is available on target router because bounce site flooded it with packets.

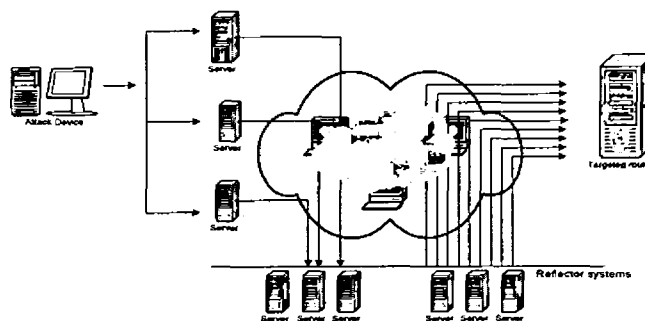


Figure 1.8: ICMP attack in Distributed Environment [19]

**Direct Distributed Denial of Service Attack:** Over the internet on a number of distributed sites Zombie software implant by attacker in DDoS. Two levels of Zombie machine involve in

DDoS attack: Master and slave Zombie. Malicious code infected both types of machines. Master Zombie triggered and coordinates by attacker which in turn coordinates and triggered the slave Zombie. Attack and its source is more difficult to trace by the use of two levels Zombie machines and attacker avail more resilient network.

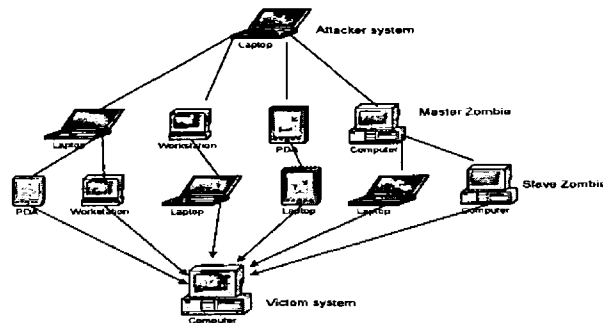


Figure 1.9: Direct Distributed Denial of Service attack[]

**Reflector DDoS Attack :** In this type of attack one more layer of machines added, and Zombie slave machines requires a response on constructing packets that contains IP address of target as IP address of source in the packets of IP header. Reflectors machines receive these packets. These machines directed these packets to target machine. This is more damaging attack because more traffic and more machines can easily involve in reflectors DDoS attack as compared to direct DDoS attack. Reflectors machines are uninfected machines so filtering out the attack packets or tracing back the attack is more difficult because widely dispersed uninfected machines involve in these attack.

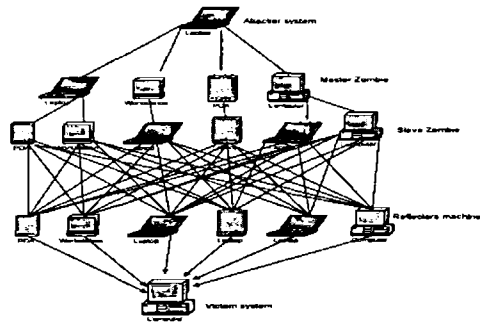


Figure 1.10: Reflector Distributed Denial of Service attack [19].

## 1.6.2 Constructing the Attack Networks

In DDoS attack the first step for the attacker is a number of machines to infect with Zombie software that will use to carry out attack.

The following ingredients are essential in this phase of attack.

1. DDoS attack can carry out by the software. By using these software attacker can communicate with a large number of machines where it run that software conceal its existence and able to launch intentional attack on the target.
2. Large number of systems vulnerable on the network. Many individual users and system administrator have not awareness of their system's vulnerability and make possibility to install the Zombie software by attacker.
3. By scanning process tracing the vulnerable machines.

In the scanning process numbers of vulnerable machines seeks out by attacker and infect them. Then Zombie software installs on them and repeats scanning process for more infected machines creation.

### 1.6.3 DDoS Attack Countermeasures

**Assault Avoidance and Preemption:** In this mechanism legitimate client avail the services while server suffers from attack. Techniques consist of implementation of policies for resource consumption and on demand provide backup resources. The chances of DDoS attacks are reduced by modifying the prevention mechanism and protocol.

**Attack Detection and Filtering:** This mechanism involves filtering out packets that occupy suspicious patterns of behavior that minimize the collision of attack on the target. When attacks are beginning this mechanism detects and react immediately.

**Attack Source Trace Back and Identification:** In this mechanism first step is trying to identify source of attack and prevent from future attack. Fast enough result does not yield from this method. By any means ongoing attacks try to mitigate.

## 1.7 Thesis Outline

Chapter1 describe overview of distributed data sharing and cryptographic techniques.

**Chapter 2** is concerned with literature survey which illustrate preceding work of researchers interrelated to our domain.

**Chapter 3** is related to problem definition and research objectives. Here we describe the scope of our research and problem domain. Next two chapters accomplish those task and goals that clearly discussed in this chapter.

**Chapter 4** presents our proposed solution and methodology. Here, we converse about our proposed mechanism according to the problem that was stated in the preceding chapter.

**Chapter 5** belongs to our result

**Chapter 6** shows the conclusion and future work.

# **Chapter 2**

## **Literature Survey**

## 2. Literature Survey

Summary of prior study on an issue is entitled literature survey. Literature review is secondary sources that aim to review critical points of current knowledge or methodological approach on a particular topic that don't account any new or distinctive experimental work. The researchers are forever exploring to discern comportment that might fetch enhancement in the existing development. As research is incredibly abundant ground, so anybody can add and contribute to his valued comprehension and knowledge, solitary at a time to prosper the technology, practice, methodologies of presently offered scheme settings. The main purpose of literature survey is to bring the reader up to date with current literature on the topic and provide the basis for another goal, such as future research that may be needed in the area. And provide the base for defining the problem domain

Greenan et al. [1] describe a POTSHARDS which is secure distributed and very long term archival storage system based on unconditionally secure secrete sharing . While other archival storage system provide a security and privacy through key encryption and key is bounded by computation effort. When data is stored for decade or centuries the use of encryption key and management of key create a real apprehension. POTSHARDS use  $A(m, n)$  unconditionally secure secrete sharing scheme that divide the object into  $n$  shares and gives refusal of information about the object until  $m$  shares collaborate with (it)  $n$  shares. The use of Two level of secrete sharing in POTSHARDS split the security and redundancy .when failure occur it allow secure restoration and more flexible storage pattern .POTSHARDS present a secure archival storage system for a despotic long period of time. POTSHARDS use data structures in such a way that an unauthorized user or malicious user would comprise a difficult time to collect shares for a particular file in proficient manner.



This data structures include build in support for consistency checking and data migration between two storage devices and rouse the reliable data.

Can et al. [2] introduces a distributed algorithm about the trustworthiness of other peers which is based on past interaction, available local information and other recommendations. Trusted third party or prior information is never used to establish trust among peers. Trustworthiness is measured in the context of reputation, service trust and recommendation trust. Three parameters are used to evaluate the interactions that are satisfaction, weight, and fading effect. Confidence and trustworthiness of recommender's is considered during evaluating recommendations. Simulation is used to understand the capabilities of proposed algorithm that mitigate the attack of file sharing application. Local information is used to mitigate the service based attack, while malicious peer achieve high reputation. In peer-to-peer system author present a cryptographic protocol on chord for anonymity of trust holder which gives anonymous access to trust information. In anonymity group each peer is a trust holder because of storing the information about others and Mental Poker Protocol is used to exchange the verification keys, by using these keys participating peers act according to the rules of anonymity protocol. Cryptographic protocol provides a security against members of anonymity group who might be an opponent. In this encryption scheme initiator of trust query can check the anonymous reply validity and responder of trust query has K-anonymity protection against opponent attack.

Subbiah et al. [3] investigates how much data can be proactively secured in practical setting? Secure storage system provide a security up to entrance of storage node. In proactive secure system time interval used to increased the security. Proactive secure system periodically cleans the state of all nodes to maintain the properties of availability, integrity and

confidentiality of the data storage service. Secret sharing or encryption-with-replication use encoding scheme to store the data at storage server. For maintaining integrity and confidentiality of large amount of data, servers periodically run the protocol. It is primary challenge in secret sharing. Author anticipated a new technique for storing data efficiently called the GridSharing framework that is a combination of XOR secret sharing and replication. Under the Mobile Adversary Model Author present a distributed protocol that periodically run by the server for maintaining the integrity and replicated data. Proactively secure document repository use a distributed protocol. Authors experimentally show both GridSharing technique and distributed protocol that can secure several hundred gigabytes of data.

Kher et al. [4] describe a comprehensive survey of the existing storage system and their security services. Critical component of corporate computing environment are storage network that evolved into complex network and distributed storage models. Number of organization growing rapidly with sensitive data (like health care records, customer records or financial data) restoring it electronically and provide online access in order to satisfy the availability and recovery requirement, this proceed the great chances of vulnerable to security breaches and damaging loses in the storage system. So this is crucial to protect such data not only in transit state but also at rest state. Author describe a number of storage system like NFS, CFS, AFS, SFS, SSFS, SiRiUS, Plutus with various security services like entity and message authentication, access control list, end-to-end data and meta-data confidentiality, and end-to-end key management etc. Some system (like NFS, SFS, TCFS) use authentication server while other system (like NASD, Plutus, EFS, CFS) use file server for authentication. Central group server performs access control in SSFS while NASD use file manager for this task. Possession of keys is used as an access control in the case of

SiRiUS. Server provide the confidentiality in AFS, NFS, NASD, and SFS system while expect these systems end-to-end data and meta-data confidentiality maintain by file server. Meta-data confidentiality is provided by CSF, and Plutus. Storage system provide another essential feature that is key recovery, EFS, CFS, and SSFS carry this feature, it is useful for recovering lost keys and old backup data. Long management key problem is solved by re-encryption of data at storage server because data exist at storage server for long time period and key should be securely managed for this long time period. An ideal secure storage system should be provided all the listed basic security services. But practically a storage system can't satisfy all the basic security services. During the evolution of storage system and quantify their requirement a system designer can take help from this list.

Hasan et al. [5] illustrates the survey of distributed file system. The popularity of these systems grows continuously due to peer to peer services like Napster, Gnutella, Kazaa, and Morpheus etc. Authors discussed various properties of peer to peer based distributed file system like symmetry, decentralization, anonymity, scalability, churn protection, load balancing. P2P philosophy is used in these file systems which are CFS, free Net, PAST, IVY, Ocean Store. Distributed file system research used P2P system concept in the development of competent file system. Now an important part of file system research is P2P distributed file system with these characteristics like decentralization, symmetry, robustness, availability, and persistence of data. The requirement of upcoming computing environment will be fulfill by these systems though there exist a limitations. So the various research activities in the field of distributed peer to peer file system discussed in this survey paper and also explore the related important issue and design paradigm.

Storer et al. [6] describe the amount of digital information is continually increasing by storing the user record and their personal histories as well as government regulation and public desires. Security mechanisms of modern systems depend on encryption in which management of keys for a long period of time is very difficult. Opponent can wait cryptanalysis technique to attain the safe and sound data by catch up the encryption algorithm which is used at the time of negotiation. Thus authors have developed an archival storage system called POTSHARDS that present a long-term security with no encryption in favor of endurance epoch data. Separate archives manage the splitting and spreading result shares to achieve the secrecy. Data recovery and availability is difficult in such system so authors present a new technique which is a conjunction of approximate pointer and secure distributed RAID technique to provide the availability and reliability across these independent archives. Prototype POTSHARDS implementation is developed to validate the design which has demonstrated normal storage and index that are used for data retrieval and recovery of user data by solitary pieces that user has stored across the archives and failed archives are completely reconstructed.

74-8875  
Storer et al. [7] describe currently there is no archival system that adequately fulfill the demand of reliability, low power, cost effective, and easy to maintain digital storage. Tape based archival system require the legacy hardware for preservation, and its random access performance is very poor that avoid auditing and inter-media redundancy technique. For long term storage many disk based system are also ill-suited for archival purpose because of management requirement and high energy demand. Authors present a solution in the form of Pergamum that is an intelligent distributed disk based network and stores the data reliably and energy efficiently by storage appliances. Idle disk save the energy in the existing MAID system whereas by the use of NVRAM at each node in Pergamum stores the data signature,

metadata request and verification of inter-disk data though the disk is powered off. To provide a protection against data loss Pergamum used both inter-disk and intra-disk redundancy and verify the correctness of stored data through hash tree like structure of algebraic signature. This approach provide a very high reliability and comparable cost in both start and ongoing to other archival storage technologies. In the case of failures, Pergamum reduces the peak energy usage by staggered reconstruct. Pergamum provides adequate performance that shows through its evolution of implementation.

Kwon et al. [8] describes the development of DHT based overlay network for routing and lookup efficiency. Kad, chord, Pastry, and CAN are representative of DHT-based overlay network. File Sharing and distributed storage system are the application of DHT based overlay network. Several security attacks like peer ID attack, message routing attacks, and rapid join/leave attack exist in the DHT-based overlay network. Authors projected a secure mechanism for DHT-based overlay network message routing attack. By using a key one peer send a look up message, mechanism ensure with high probability the message is delivered to the owner of the key peer. Recursion hash function and lookup delegation are two mechanism used for this purpose. Lookup success ratio increase by this mechanism even malicious peer nodes exist on this network. Authors provide simulation results of this mechanism that shows the highest look up success ratio then the original P2P overlay routing protocol, chord.

Storer et al. [9] describe the characteristics like integrity authentication and privacy threat of long term storage that does not exist in non archival storage system. Long-term storage system and set of threat might lead to system compromise. Authors have designed write once, read may be usage model for long term archival storage system and set of threat. They just present existing and new threats which are specific to the long term storage system

rather provide a solution of these threats. Authors have examined the survivability of long term data and its threat in the existing system which are Free Net, Ocean Store, Far Site, PAST, Publius, SNAD/Plutus that provide secrecy through encryption while Grid Sharing, PASIS, Clever Safe, POTSHARDS provide a secrecy through secrete sharing and LOCKSS Glacier, Venti have no specific secrecy mechanism. Ocean Store provide authorization by signature, FarSite by certificate PAST by smart card, Publius by password and SNAD/Pultus by encryption POTSHARDS by pluggable and Glacier by node auth and Venti, LOCKSS, PASIS, Grid Sharing still not define the authorization. but all the existing storage system provide integrity by using different mechanism, none of them addressed the slow attack, Free Net, Ocean Store, Far Site and LOCKSS offer a migration facility. Hence none of the system that address all the security concerns and existing threat. slow attacks must be addressed in future because it is a great risk for long term archival storage system. Long term storage system will be accommodating hardware changes and immigration. Absolute and more focused secure archival storage system should be build in future.

Endo et al. [10] anticipated a distributed storage system which is based on self-encryption scheme, the core endeavor of this distributed storage system is to protect stored confidential information of mobile handsets. Authors describe a self encryption scheme or certain algorithm that use target file to produce an encryption key. The information is divided into two pieces in this proposed system, one piece is stored at mobile handset local storage while other piece of information is pile up at network storage by uploading the information through a safe and sound path which is provided by IMS. By using self encryption scheme uploaded plain text file is encrypted into s distributed data for the relaxation of load of network server and the mobile handsets. By hacking a single server of sharing network storage infrastructure with other client which is based on P2P overlay network the intruder can't acquire all

distributed data. Uploading time is reduced by omitting the physical upload procedure if other users already stored the data by uploading. High scalability proficient fault-resilience provided by super node P2P network storage, application server use index that is provided by IMS. But for the standardization of this distributed storage system following protocols is essential like server and mobile terminal authentication protocol i.e. IP Multimedia Service Identity Module (ISIM) for mobile terminal authentication and Public Key Infrastructure (PKI) are used to server authentication). For communication self-encryption scheme algorithm and protocols are required and also originate DHT IDs method.

Bathee et al. [11] proposed a mechanism in which system requirement of bandwidth is reduced by distribution and decentralization of block generation. In this scheme any node can randomly join and leave the distributed storage system in the storage network. Reliability is ensured by the redundancy when data is stored across the set of distributed nodes. Reliability of stored data is the main purpose of distributed storage system. Erasure code is well known redundancy method maximum distance separable and rate less codes are the example of erasure code usually used for the purpose of redundancy. Storage nodes can randomly join and leave the system in a distributed storage environment that's why called dynamic distributed storage system and in time refreshment of redundancy is necessary. The entire file i.e. all original blocks availability is ensured by regenerating new blocks through known erasure code. For the construction of one encoded block the new storage system should download the complete file from that one system where replica of original file is always available both cases are not appropriate because of across the network large data transfer and the need of very reliable nodes that may be one or more. In the network coding area author proposed a mechanism for distributed storage system. According to this proposed scheme new storage node require a downloading data equal to the block size for the construction of

new encoded block so redundancy is acceptable by maintain the required bandwidth and reliable nodes does not require for keeping the replica of original file. With high probability data collector can obtain the original file from the set of arbitrary storage nodes by downloading the file size equitant data. So in retrieving the original file and redundancy maintaining in both cases bandwidth is efficient according to the proposed mechanism.

Dimakis et al. [12] introduces three schemes in this paper for peer to peer distributed storage system in which two new scheme presents a redundancy maintaining by using erasure code and one is general technique that combine any form of replication and coding to analyze storage architecture. First describe in the system maximum distance separable (MDS) fragments directly optimally generate from existing fragments. Secondly introduce slightly larger fragments than MDS fragments called regenerating codes that used a lower bandwidth overall. Across the internet through redundancy data is accessed reliably over spread nodes in peer to peer distributed storage system redundancy is used to minimize the bandwidth. Erasure code is used to store a file in fragments that spread across the nodes and less redundancy and less bandwidth maintenance is required then same level of reliability is provided by simple replication authors demonstrate through simulation 25% or more compared with the best previous design in a realistic environment regenerating codes can reduce bandwidth maintenance.

Ylianttila et al. [13] describe both fixed network and in distributed mobiles need the data management in efficient and scalable manners. In peer to peer scenario especially considering enterprise servers, mobile devices and personal computers lies in heterogeneous and distributed, autonomous network that share stored multimedia data, amount of this data is growing rapidly. In this paper authors present a simple but extensible analytical model that



compare the emerging distributed data management system with existing data management system with various aspect. Internal and external parameters of the system used to analyze the service efficiency at indicating operational cost. In mobile P2P environment design guidelines consider for data management and support optimal design choices, model is applied to benchmark distributed data management. Distributed data management system that based on P2P system, first generation centralized client server system and second generation distributed client server system these are different data management solutions. Author analyzed in this paper by viewing the features of mobile environment in the existing system. An evaluation framework is introduced in this paper through this schema.

Serhani et al. [14] describe in this paper about extensive number of users explore online services and use number of mobile applications. Actual handheld devices have a considerable computing power but the utilization context is different. Hard keyboard or on screen, size of screen, battery life, wireless network with high latency context are affected by the connection availability, high quality service will provided by a concise methodology associated mobile web service and development of mobile application limited resources, security and communication related other challenges need a well defined development environment which handle them. Therefore author of this paper presented a new frame work that leads developers to develop a secure and efficient application through a different steps and modules, first elicited and deeply discussed such applications face a different challenge in developing. Second these challenges addressing a different modules is presented by framework development.

## 2.1 Limitations of the Literature Surveyed

In [1, 6] these papers long term archival storage system infrastructure is implemented without using encryption scheme but still some outstanding issues. This archival relatively reliable but challenges is remain for automated malignance of large scale archival storage system. Changes will certainly occur in basic hardware so storage survive from decades to century becomes a critical issue for example currently tape storage is replaced with power managed disk array, Currently potshards keeps the data safe by using the strong authentication and intrusion detection scheme but detection of intrusion after many years is not clear. Refactoring the data approaches are available and partial progress in an intrusion can be cared by making new shards incompatible with old shards, slow attack could succeed on refactoring security.

In [2] this paper by using local information can reduce the service based attack, recommendation based attack are not mitigated. Malicious peer are able to gain high reputation. An adversary can be a member of anonymity group and can launch active attacks.

In [7] this paper network connection per disk and CPU optimality is also an open question our choice is based on both quantitative and qualitative but other arrangement certainly possible, additionally modern desktop level CPU can used at client side machine that could be influenced for pre-processing and also consider an interesting problem to determine the best network to use to contact thousands of (mostly idle) devices. Securely and power efficient manner for data migration is another requirement. In long term reliability strategy the basic part could be a device refreshment policy and implementation for reliability can also open research direction. Power consumption and storage overhead and the interplay of

redundancy understanding there is still work to be done. With a member of interesting problems archival storage in general and storage management in Pergamum is an open area.

In [5] this paper mostly P2P file system not actual system deployment. These are designed as an academic research projects so analysis the performance of these systems by controlled data simulation. In a real world situation only the freenet system has been deployed.

For most data intensive application the requirement of high bandwidth never fulfill by the best current generation of the P2P file system. In most systems the mechanism for assigning globally unique file identifier is ambiguous. Those systems that use content hash key the key will be changed if small amount of data changed. Churns that created by malicious users will degrade the performance of the system location mechanism and hamper system convergence.

There is very little mechanism that ensures the authenticity of mechanism. On P2P host distributed denial of service attack can be made. Overall performance reducing by brings in junk files into system from the malicious peer posing as a valid peer. Encryption provide a safeguard but if malicious user have access to the data then dictionary based and brute force offline attack can't stop by this scheme. In distributed peer to peer system inherently there is no administration mechanism so administrative policies can't be enforced.

In [3] this paper 100 GB of data can scale on the emulab testbed as measured for proactively secure system, other system can proactively secure if the periodic refresh mechanism applied on these systems. As a good system administrative proactive can be runaway by such mechanism. Perfect secrete sharing scheme is an efficient technique for storing data provided by a Gridsharing framework. If XOR secret sharing replaced by generic  $(n, n)$  linear secret sharing schemes then distributed computation performed over stored data over their shares, the use of replication with voting by using the distributed computation not made be verifiable.

In [10] Self encryption scheme: the performance of the system can be slow down because look up load on P2P network is large. That's why super node is required. Server does not have proper authentication mechanism so mobile devices collectively can launch flooding attack on the server. Data send from mobile devices to server is conveyed in plain text sending sensitive data in plain text raise many security threats. It is fixed only 50% data can be transferred to the server; no valid reason is provided about this value. Devices are not allowed to send 49% or 51% data.

[13] Authors used educated guess for calculating operational cost and grades while in future for defining these cost and grades improved methods will used. Both factors defined by using real life application scenario and developing practical.

## 2.2 Summary

Communication mode of the people has radically changed seeing as the communication of early days. Dwelling phones, electrical and optical telegraph drove the insurrection from the prior combustion signals by user's requirement. Mobile terminals like PHS(Personal Handy-phone System), cellular phone, note PCs etc are simply carried out and they are forever on that's why users of mobile terminals boost up day by day, these terminals are extensively used for storing the business information . There are certain critical issue regarding shares a common data among PC and cellular phone and PHS then face a trouble. The prior storage systems presented by various authors in this chapter are not much enough to solve such problems, but this type of storage services is not preferable or suitable for the treatment of business information. It entails such proficient safety measures which are able to make sure the stiff security and proper mechanism for data sharing.

# **Chapter 3**

## **Requirements Analysis**

### 3. Requirements Analysis

Requirement analysis is a method of perceptive requirements of the users, in contemporary circumstances evaluating the problems, about the system congregating information, and accomplishes needs of the end users via endeavor to solve the problem by evolving new technologies. For accomplish the system's and system user's expectations the requirement analysis will oblige the developer to commence innovative dynamic alteration in the present network. Currently wireless communication is a part of our daily life or a global community. In this form of technology a lot of security and privacy issues arises. Now a days mobile security and privacy is a central issue in a society have a number of factors and their additional features. These mobile terminals are merely carried out and they are forever on therefore users of mobile terminals increase day by day. Now these terminals are commonly used for storing business information. Users store the data records at these terminals in a dissimilar format but due to dissimilar data format they face difficulties for sharing a common data format among PC and cellular phones and PHS. Distributed storage system is anticipated for defensive confidential information stored in mobile handsets in 2008. By using self encryption scheme information is distributed into two parts in this storage system one part is at the storage of mobile terminals and another part is stored at the network storage. Users make effective use by selecting the file splitting algorithm while mobile terminals have limited resources.

#### 3.1 Problem Definition

After conducting extensive literature survey now we are going to formulate the problem definition. Battery limitation is one of the main problems in the current mobile age. Since mobile devices have limited battery so encryption and compression is not preferred at mobile devices. Researchers developed a solution to store the contacts at different servers. [10].

Mobile devices send half of their contacts in plain text to different servers. Servers after receiving the contacts encrypt them and store them. Whenever the device requires those contacts it request the servers and server decrypts the contact and send it back to the device.

In this solution first of all no proper authentication mechanism is described so the mobile devices collectively can launch flooding attack on the server in the absence of authentication mechanism.

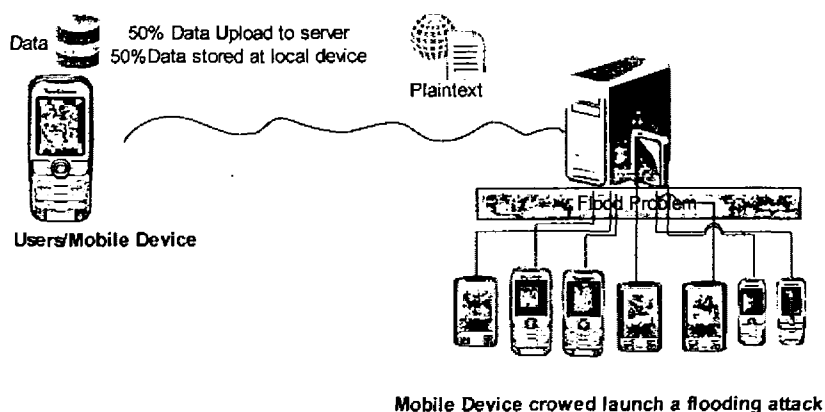


Figure 3.1: Flooding Problem at Server

Server tries to entertain all launched requests that contain not only legitimate request but also spoofed request due to absence of authentication mechanism. Spoofed request is also entertained by the server and allows the risk of flooding attack as well legitimate users suffer a lot because they can't access the resources properly from server.

Secondly data from mobile device to server is conveyed in plain text. Sending sensitive data in plain text raise many security threats.

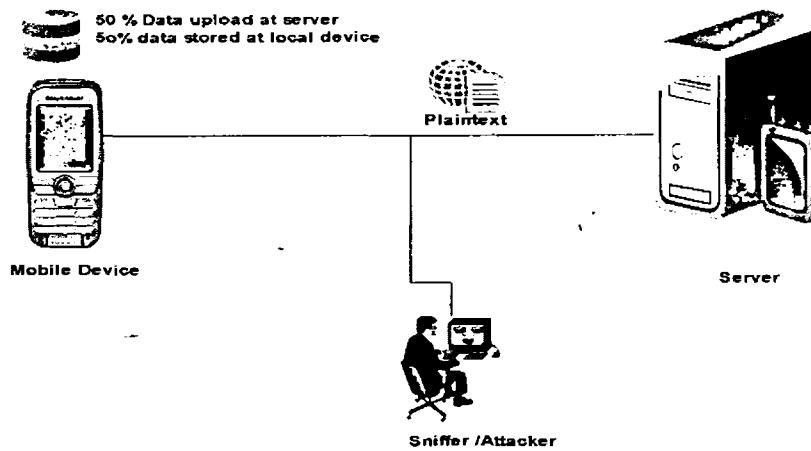


Figure 3.2: Plaintext Conveyed to the Server

When data is transferred in the form of plaintext from mobile device to server it will raise unsatisfaction because of any opponent can easily sniff or get the half confidential information and he can easily understand the message.

Thirdly it is fixed that only 50% data can be transferred to the server-No valid reason is provided about this value that creates adaptability problem. Device is not allowed to send 49% or 51% data.

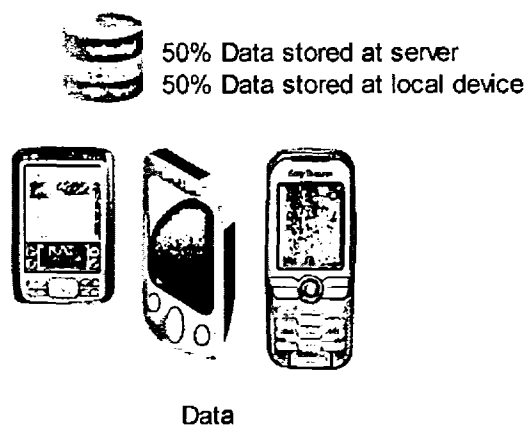


Figure 3.3: Mobile Device Capacity



The third part of the problem definition is related to division of data and information. According to authors techniques data or information is always divided into two equal halves in every type of mobile devices instead of capacity of mobile devices. If any one user don't want to upload 50% or half data or he want to upload more than 50% data because his mobile capacity is low or any one user want to upload less than 50% of its confidential data because his mobile capacity is elevated Then these types of users have not option or relaxation for uploading the data at remote site or server and at the local storage of the mobile handsets according to their choice.

## **3.2 Research Objectives**

Proposed approach of any solution is based on the objective of research. Aspire of research correspond to the objective of research which is conducted for problem resolving. The origin of solution design is the objectives of research and each problem is resolving by manipulative its resolution. Beyond discussed problem statement we have situate a few objectives on the basis of these problems. The objective of this research work is to reduce the security threat during transferring the confidential data or contacts from mobile phones to servers, because transferring the data in the form of plaintext has a lot of threats. Anyone can easily get half of the confidential information. This research work also helpful to mitigate the flooding attack on severs because we have introduce authentication mechanism on the server and also provides a dynamic division capability that provides ease to the user from data sharing point of view.

### **3.2.1 Utilization of Resources**

In our proposed solution the resources can be utilized efficiently and effectively.

### **3.2.2 Mitigate Denial of Service**

When requests are served according to the queue on the server as a result of there is no denial of service. Some scheduling mechanism applied for serving the entire request.

### **3.2.3 Cost Effective and Enhanced Applicability**

The proposed solution is supplementary cost effective. It augments the applicability of the contact number or data sharing services in daily life. It provides control information to the users and also enhances the scalability of data sharing services by allocating different quota rights to the users.

### **3.2.4 Minimize the Chances of Flooding Attack**

Our strategy will facilitate only authentic users. Only legitimate users can interact with server and unauthentic users can't access the server so our solution minimizes the risk of flooding attack on the server and that's how our legitimate users get better service.

### **3.2.5 Minimize the Loss**

One of the core objectives of our research work is to reduce the loss which is earned by spoofed users or sniffer. A solution which we present in this dissertation will provide partially security to the data from opponent if the opponent gets the data he can't understand the message and can't easily decrypt the part of confidential data.

### **3.2.6 Provide an Efficient Service**

To make available an efficient and effective service to the users is the major objective of this research thesis. We have designed such a mechanism which mitigates the security threat for mobile device during transferring their contacts to the server and server will save from flooding risk by applying a proper authentication mechanism on the server.

### **3.2.7 Increasing the Overall Data Sharing Security**

To overall data sharing security is increased by applying the security means during transferring the data from mobile phone to server. The main objective of this research is to

offer or provide an absolute security to the users during communication. If the adversary gets the data he can't understand the message and cannot easily decrypt the part of confidential data. Hence this mechanism will add the overall data sharing security.

### 3.2.8 Evaluation and Formulation:

Every strategy or technique is devised and evaluated.

## 3.4 Summary

Mobile phones become a part of our daily life and users of mobile terminals boost up day by day. These mobile phones are extensively used for storing the business information. At these terminals users stores the data in different format and face a trouble due to different data format when they share a common data among PC and cellular phone. So, researchers commence the distributed storage system for defensive confidential information stored in mobile handsets. Mobile devices transfer their contacts in plaintext to different servers, after receiving the contacts servers encrypt and stored them when device need those contacts it request the server and sever decrypts the contacts and send it back to the mobile device. In this solution mobile devices can cooperatively commence flooding attack on the server in the absence of authentication mechanism. Sending sensitive data in plain text heave numerous security threats. Devices are fixed only 50% data can be transferred to the server and can't avail dynamic division. We set numerous goals and task in the research objective. We provide an idea of secure data sharing from mobile phone to server by applying authentication mechanism on sever; only authentic users avail the services of server. Applying some scheduling mechanism for serving the entire requests provide. Present dynamic division of data and apply some security measures on the plain text during transferring the data from mobile device to server.

# **Chapter 4**

## **Proposed Solution and Methodology**

## 4. Proposed Solution and Methodology

The problem definition restrain data sharing technique and issue which we were facing during that data sharing from mobile phone to server, data from mobile phone to server is conveyed in plaintext, in existing system deployed authentication mechanism at server does not have a proper mechanism to separate spoofed users from legitimate users and users are restricted at 50% data can be transferred to the server. No valid reason is provided about this value. Our proposed solution excel the research objective summarize in the previous chapter [section 3.3]. By considering the problem definition we grasp to design such solution that can be sufficient to cope with the problematic situation. In this chapter we have proposed the strategies for data sharing from mobile phone to server applying a proper mechanism on server and also provide a reinforced security dimension during transferring the data and also provide ease to the users for uploading contacts division. The solution we presented in this research thesis is capable to accomplish all those goals and objectives which we have discussed in previous chapter. Here we have introduced a new data sharing technique which protects the server from flooding risk and servers can properly response to the legitimate users. This technique provides a security to transferring the confidential data from mobile phone to server by applying some security mechanism. Proposed technique is more efficient and works more immaculate than the earlier data sharing techniques. Data is transferring in the form of plain text from mobile device to server that produce un-satisfaction because of any adversary or sniffer can easily sniff or obtain the partly confidential information. Our proposed solution resolves this problem and provides a security by encryption for transferring the confidential data from mobile device to server. By applying some security mechanism or present a partial security to the data from adversary if the adversary gets the data he can't recognize the message and can't decrypt effortlessly that part of the confidential data.

Therefore we can say that our proposed solution is not compromising on the confidentiality and privacy of the users.

Our proposed solution also provides a dynamic division of confidential data. It should depend on mobile phone device capacity it can be 40/60 ratio or according to user's choice. There is no restriction to divide the data into two equal halves. In this research thesis we have designed an adaptive data sharing technique between mobile phones and network storage and applying some security mechanism during data transferring. Numerous techniques those proved efficient in one scenario give a pitiable result in another scenario. So our proposed solution provides a proper mechanism for data sharing from mobile phones to server. These capabilities eliminate the flooding risk from server and provide a security for transferring the confidential data. In this research thesis we will design a new private distributed storage system technique. This new storage system technique will work on mobile device according to its capacity that connects to the network storage or server for uploading contacts. A lot of research papers have written on distributed storage system that proposed an efficient file sharing infrastructure. Several techniques those proved efficient in solitary state give a poor result in another state of affairs. We applied an authentication mechanism on the server when an existing user send the contact uploading request to the server, the server verify the username and password from the database, if the user/mobile device is authorized then the server allow him to upload the contact according to assigning quota in the server database. If the username and password is not valid then the server rejects the request, this also enhances the through put of the system.

Our anticipated solution apply a appropriate mechanism on the server when verification of user by username and password is complete and confirm the user is authentic then user

request for uploading data to the server among the information of mobile model, battery power, storage space and percentage of data records which would like to upload, server received the request and verify the current status of mobile device and calculate the percentage of data which permit to the user for uploading and show a response to the user with upload id also percentage of calculated data, in existing system data is transferred in the form of plaintext from mobile device to network storage or server creates un-satisfaction because of any opponent or sniffer can easily sniff or get the half confidential information. Our proposed solution has provided a security to transferring the confidential data from mobile device to server by applying some security mechanism. We have secured the data or information during transformation or uploading by providing a partially security to the data from opponent if the opponent get the data he can't understand the message and can't easily decrypt the part of the confidential data. Hence data record is encrypted by encryption algorithm at mobile device, as server is free it pull the data record through upload id, at that moment user upload the data record at server, and server acknowledges to the user about uploaded data. By this mechanism just legitimate users are able to interact with server and non authentic user cannot access the server therefore our server saves from flooding risk. Thus server is capable of well response towards the legitimate users.

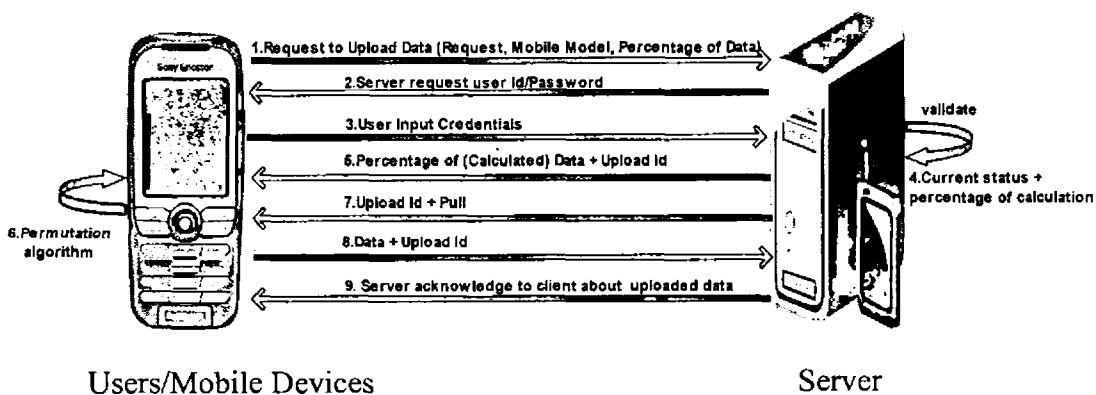


Figure 4.1: Proposed Solution

And the solution of third part of the problem definition is author always divide the data into two equal halves or use a fix technique while we have developed an adaptive data sharing technique that decided when and how much data should be transferred that depends on battery power and storage capacity of mobile device. In this technique dynamic division of confidential data depends on mobile phone capacity that can be 40/60 ratio or according to user's choice. If our mobile phone capacity is fewer than minimum data stored at mobile device and larger part of data upload at server and if we have high capacity mobile device we can store large part of data at mobile device and less part of data upload at server. So users can feel ease according to his choice there is no restriction to divide the data into two equal halves or fixed amount of data.

Now we are presenting the detailed functionality of our proposed solution.

## **4.1 Authentication and Data Sharing**

A proper authentication scheme is required to transfer the data from mobile to server. In our solution authentication is also performed at client side. When end user wants to logon to the client system, client system check its local database before granting access, database ask users credentials that are usually username and password when user provide these credentials for authentication, after receiving these credentials client system verifies the password and username from its local database. Upon the validation of local database, system granting the user to access the client system and if these credentials pair does not match with existing database then refused for granting access.



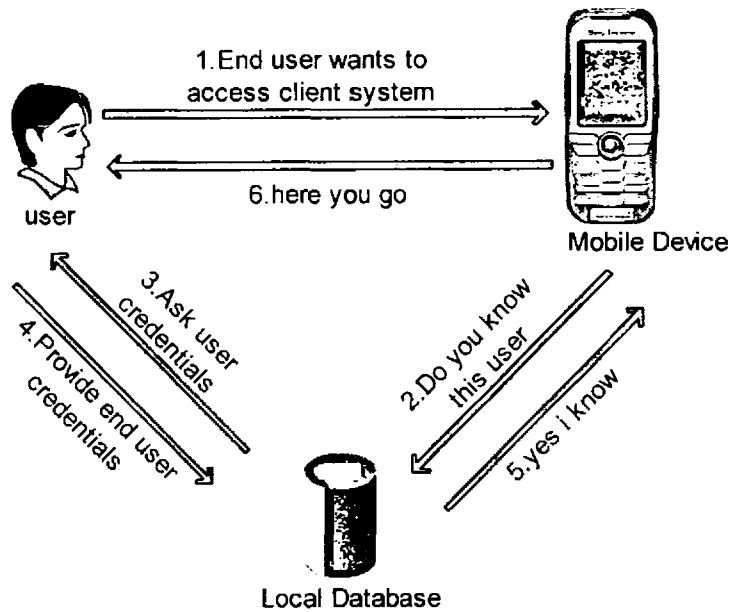


Figure 4.2: Step 1: End Users Get Access Client System

Following algorithm shows the user authentication:

```

V_username, V_password, V_passkey, V_response;
Begin
V_username = get_user ();
V_passkeyn = get_passkey();
V_password = get_encrypted_data (get_password (), v_passkey);
V_response = valid_user (V_username, V_password);
Message ("V_response");
Exit
  
```

When user get access to the client system, he can add data, update existing data in the database and also can delete the data from database.

Aim toward standardize access control by authentication and authorization mechanism, for the reason that many servers contain sensitive information and just authorized users have

access to it. Usually access is controlled by authentication procedures that allocates user identity and then grant those privileges that authorized to that identity. Usually username and password are used as identification credentials. A person is considered to be an authentic user if specified username and password is valid and after that determine the resources to users can access. When end users enter into client system and if he wants to upload data then client initiate a data upload request to the server. When client initiate a data upload request as well initiate a TCP connection request by sending SYN packet to the server.

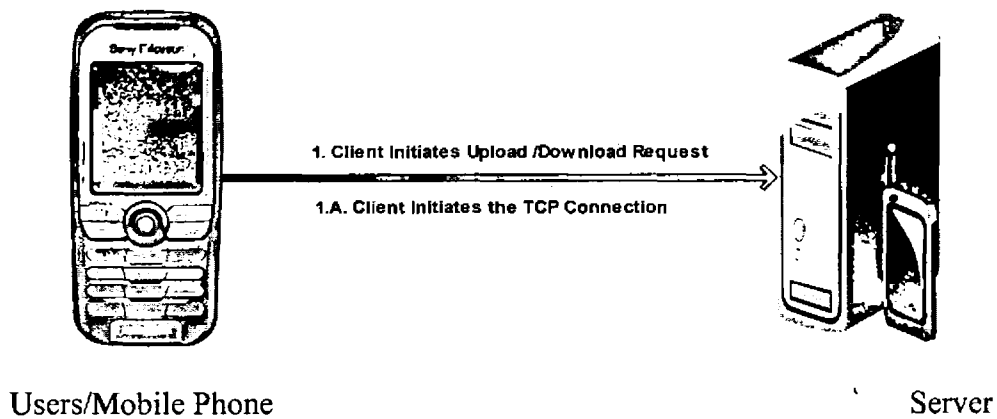


Figure 4.3: Step 2: Client initiated TCP connection and data upload request

TCP is connection oriented transport layer protocol that guaranteed data delivery. TCP connection is established by three way hand shaking between client and server. In first step our client sends a data uploading request to the server as well as initiates TCP connection by sending SYN packet to the server. If there is no good authentication scheme then a flooding attack may be launched by sending malicious data transfers. To stop the malicious user whenever a server receives a request from a mobile phone it authenticates the mobile phone by sending a user name and password. Mobile user needs to return that user name and password to the server. If the request arrives from a spoofed user then it will not be able to

return back the sent user name and password. So by sending user name and password we can authenticate that whether the user is spoofed or not.

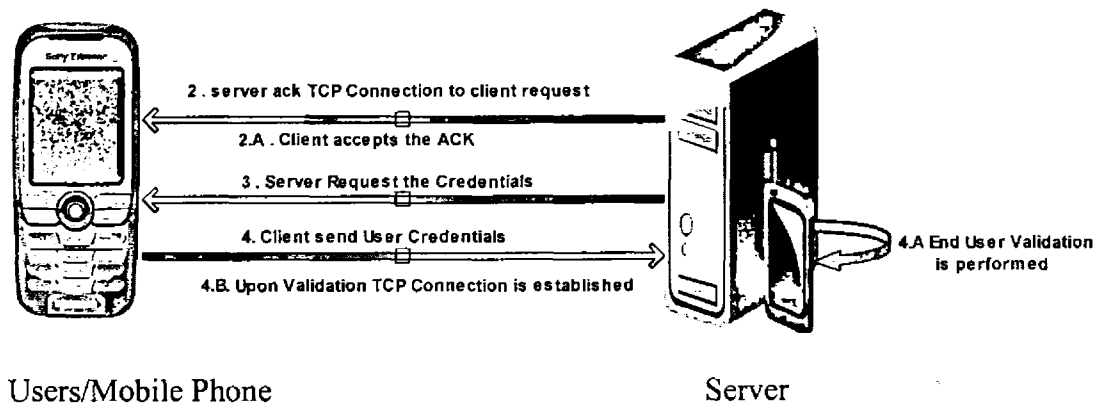


Figure 4.4: Step 3: TCP connection established

Server receives the SYN packet from the client and also uploading request then in the response of sends SYN/ACK packet to the client and as well ask the user credentials to the client for the authentication of users. This authentication performs at application layer while TCP connection is established at transport layer. Client provides the credentials to the server for validity. Server received these credentials and performs the end user validation. Server checks its database for the validation of end users and upon that one validation TCP connection is established. If end users credentials are wrong then it is considered as spoofed user and server refuses its connection establishment. As a result server entertains only legitimate users and provide them proper services. By refusing the spoofed or illegitimate users the chances of flooding attack on server is reduced as well saves from insufficient bandwidth problem. By applying proper authentication mechanism spoofed users initiated TCP connection are refused and as a result round off all initiated legitimate TCP connections are established which improve the system performance.

After confirming that user is not a spoofed one there exist another security threat that is known as Zombie clients. There can be few machines those are compromised in such a way that whatever it receives from the server it returns the same back to the server. So a zombie machine whenever receives a user name and password from the server it returns it back to the server and get authentication. To block the zombie machine image based authentication like CAPTCHA can be used. Most of the zombie machines will not be able to read the zigzag text from the image so threat of zombie machine will be reduced. After authenticating the users we are not allowing it to share the data because it can results in overloading the server.

## 4.2 Calculation of Amount of Data Transferred

When TCP connection is established between client and server then initiated data upload request from client is entertained by server. In existing system the devices share its 50% data on the server to improve the storage space. Sharing exactly 50% data for each and every mobile phone is not feasible because mobiles have different storage space, different processing speeds and many other different parameters as shown in the figure below.

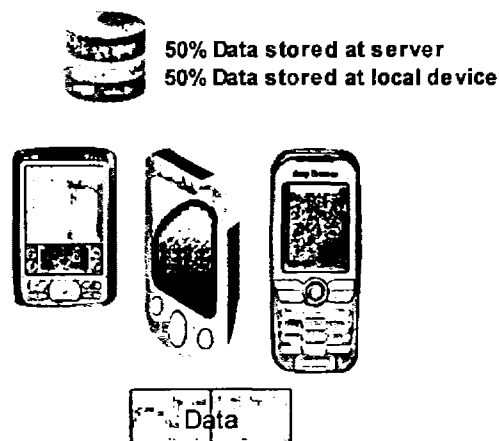


Figure 4.5: Mobile Device Capacity

Instead of sharing exactly 50% data from each and every mobile an adaptive system should be prepared that calculates the amount of data needs to be shared on the server. This amount will vary from mobile to mobile. In our proposed scenario whenever a mobile sends a request to upload the data on server it also sends few of its parameters like processing speed, storage space, software supported etc. Upon receiving these entire parameters server calculates that how much data should be allowed to upload.

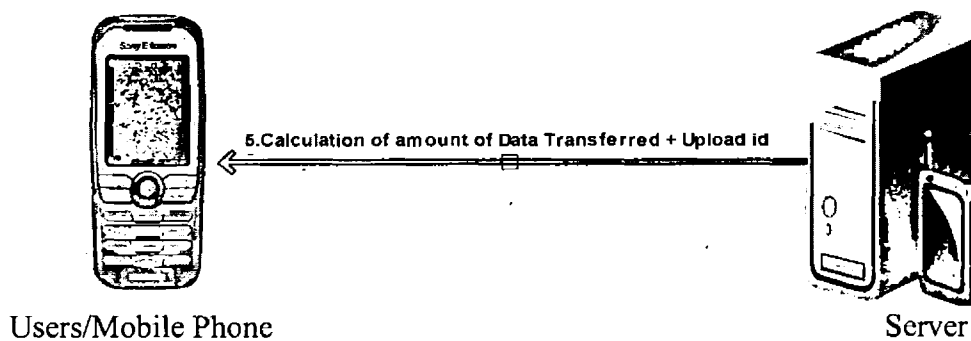


Figure 4.6: Step 4: Data is uploaded at server

We designed a formula that is used to calculate the amount of data shared by each device. Server check request parameters (i.e storage space and battery power) then server calculates what portion of data should be uploaded on server, the amount of data is not fixed , it is decided on run time that how much data should be received from client end on the bases of storage space and battery power . Data uploading ratio vary on the same battery power because of less storage space and sufficient storage space classification. Following formula is used for taking this decision. In which we assign a constant value 1 for less storage space and 1.2 for sufficient storage space and then by using the following formula we calculate the amount of data that is uploaded at the server from mobile device.

Begin

Less storage space is assigned value =1

Sufficient storage space is assigned value = 1.2

Formula is :

if battery power < 500

Total uploaded data is = (Total battery power /20) \* Storage Space

if battery power < 1000

Total uploaded data is = (Total battery power /32) \* Storage Space

if battery power < 2000

Total uploaded data is = (Total battery power /40) \* Storage Space

if battery power < 3000

Total uploaded data is = (Total battery power /50) \* Storage Space

if battery power > 3000

Total uploaded data is = (Total battery power /70) \* Storage Space

Exit

Total storage space will show that how much data mobile can accommodate within its internal memory; it will help to calculate the total allowed data to be shared. Mobile will also inform the server that how much data it wants to store on the server. This parameter will help to decide the server that whether it is feasible for the mobile phone to share this amount of data. Battery life and processing speed are two main parameters. Battery life is involved because the data storage at server requires valuable energy and accessing back the data from

the server also results in use of battery life so if the battery life of the mobile phone is less, lesser data will be allowed to share and if the battery power is more, more data will be allowed to share. Processing speed of the mobile is also an important parameter because sending and accessing data to and from server requires some processing and if the processing power of the mobile is low lesser amount of data will be allowed to share and if the processing power of the mobile is higher, more data can be shared by mobile on the server. Based on above mentioned formulas, server calculates how much data should be received from client side and saves the battery power for communication. Our solution is adaptive. it adjust the shared data according to the available battery. There is no issue of less data sharing and more data sharing. Sever shows a response to the client about its availability.

### **4.3 Securing the Data in Transit from Mobile Phone to Server**

In existing system whenever a mobile phone is allowed to transfer data to the server, it sends the data in plain text. Any sniffer can be used to capture the data and since this is private data of the user so capturing the data in the transit raises many privacy and security issues. To somehow secure the data in transit we add up a security step before transferring the data from mobile to server. Efficient authentication protocols like DES and RSA can not be applied in this scenario because of two reasons. First the key supported by most of the mobile devices is maximum of 48 bits so DES and RSA can not be used in this scenario and secondly processing cost of algorithms like DES and RSA is so much that mobile phone's battery will expire in encryption process. So we need a solution that on one side secure the data and on the other side consume lesser amount of resources. We applied a simple permutation based algorithm to somehow secure the data in transit. Whenever the mobile phone has some data to transfer to the server it divides that data into different chunks. Say if the mobile phone has 10 MB data to send to the server it converts it into 20 chunks and after making the parts

mobile phone applies a simple permutation based shuffling algorithm to mix the data and to make it un-understandable.

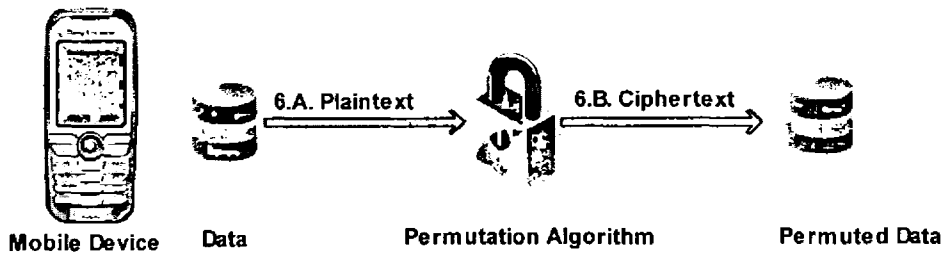


Figure 4.7: Step 5: Permuted Data

Now when data is transferred from mobile to server only very intelligent hackers will be able to get its actual meanings. Ordinary or routine hackers will not be able to get the meaningful data because it is shuffled and bytes are not in proper order. We are not using any popular encryption algorithm, in cryptography change in order of unit or according to regular system the position held by unit of plaintext are shifted is permutation of plaintext characterize the cipher text that is a method of encryption called transposition cipher. We are just applying simple permutation on the chunks of data and for this 48-bit key is used. We are not replacing data with other data; we are just shuffling data so it will consume the same space as the original data will take. So there's no effect of consumption of space by applying permutation algorithm. We are just using simple permutation based on shuffling technique. In ciphertext transposition moves the plaintext to unpredictable places and somewhat difficult to analyze it, any permutation algorithm can be used for it. Transposition cipher is absolutely appropriate for short messages and quite enough for our system because we only share the contacts numbers from mobile device to server. The easiest one transposition is columnar transposition in which characters are rearranged into characters. Encipherment/Decipherment complexity of algorithm is constant per character, in the amount of work, time complexity is



$O(n)$  in this time is proportional to message length, space complexity is  $O(n)$  and requirement of space is directly proportional to message length, delay is also depend on message length that is  $O(n)$ , and all characters are readout for the production of output characters.

Algorithm for data encryption:

```
Declare
V_f, V_data, V_Passkey, V_Edata, V_response;
Begin
V_data = get data ();
V_passkey = get passkey ();
V_Edata = get encrypted data (v_data, v_passkey);
If ( v_f = 'I' )
Then v_response = add data (v_Edata);
Else
If (v_f = 'U' )
V_response = update data (v_Edata);
Else
Message ("Not valid action");
End if
Message (v_response);
Exit
```

Traditionally pull and push mechanism used for data delivery, in which server require to push the updates to client is called push mechanism, and in which clients needs to check for

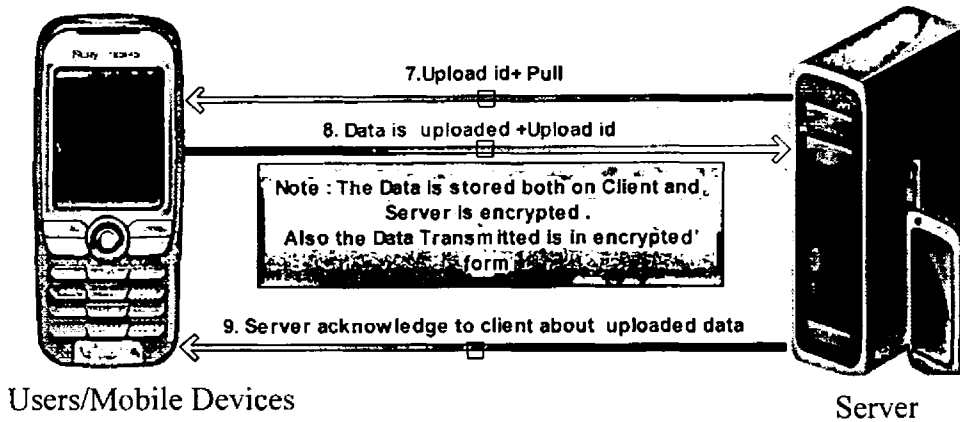


Figure 4.8: Step 6: Data is uploaded at server

In which when a user or mobile device send a request to the server for uploading data, then server after authentication and verification of the users, Server checks its load and whenever it finds it feasible it sends a pull request alongside the upload ID (issued after the authentication) to the mobile phone. On receiving the upload ID the mobile phone verifies it and start transferring data on server. This pull based technique allows the server to demand the data according to its own availability. So chances of overloading the server through bulk of data transfers are reduced in this way. The server always keeps the load balancing algorithm in running form and asks the mobile phone to transfer data accordingly. Pull the data from mobile device that is saved in ciphertext form in its local database. Server pulls the data packets on selective response to demand, server response to the mobile device. When the user adds the data on client the data is stored in encrypted form by applying the permutation algorithm. Client uploaded the selected data at the server which end user wants to upload. The data that is uploaded at the server is in encrypted form that provided the security during transferring the data from client to server. If any sniffer sniffs the data then he can't understand the data or if any opponent obtains the data he cannot recognize the message and cannot easily decrypt the part of confidential data.

At the server end data is also saved in encrypted form, if server is compromised then nobody can easily understand the encrypted data and would not be able to decrypt that data. When the End user has to see the data, first the data is decrypted and then shown to the end-user on client. By applying pull based mechanism in our system, at the server irrelevant data will not arrive, and only relevant data is disseminated when server ask for it. We adopt pull based mechanism that is best one option for our server to respond many uploaded request devices within expected time period and has very little contention. By using pull strategy we save network bandwidth that is particularly important in wireless environment. Minimizing the number of contacts between clients and server for effective pull based data delivery while client's requirements satisfying. While excessive client request increase the overload on server and add latency to the client request when contacting to the server. When mobile device uploaded the data at server, then server acknowledge to the client about uploaded data that is successfully uploaded or cannot uploaded successfully.

There is another situation, when the server is not available, the data which is to be uploaded by the client is added to a queue and client keeps the queue until the selected data is uploaded. Client keeps listening for the availability of Server When the server becomes available, Client sends the request for uploading operation to the Server. In response the Server requests for end user credentials for validation. On successful Validation the queue is processed and selected data is uploaded at the server and server acknowledge to the client about uploaded data.

## 4.4 Summary

In this chapter we have presented an efficient and secure mechanism for data sharing from mobile phones to server. Our proposed solution have provided a security to transferring the confidential data from mobile device to server by applying some security mechanism because sending sensitive data in plain text raise many security threats. In this solution for the ease of mobile user we developed adaptive approach for data sharing. And by applying proper authentication/authorization mechanism on the server, only legitimate users can interact with server which saves our server from flooding risk and legitimate users properly entertained by the server.

# **Chapter 5**

## **Experiments and Results**

## 5. Results

In this chapter we are evaluating our proposed strategies in the previous chapter. To validate the effectiveness of the proposed solution we perform numeral experiments. All of the experiments present better results than the existing values. We can easily and effortlessly analyze the performance of our designed strategies.

### 5.1 Scenario 1: Data Transfer Rate vs. Battery Lives (Sufficient Storage)

In this scenario we took 5 clients with the battery power 200 Joules Poor, 800 Joules Fair, 2000 Joules Good, 3000 Joules Very Good and 5000 Joules Excellent respectively. Now the capacity of each client is raised to 500 Mb. The size of queue is 100Mb. The connection between client & server is TCP. The existing solution does not care about the Battery utilization on the other hand in our proposed approach which is adaptive in nature cares about the battery consumption to keep the client live.

In this experiment, we have measured battery power against data transfer rate; we take the existing system results as it is available. We run our proposed mechanism and test the data transfer rate with different battery powers. A mobile node that has sufficient storage space and 800 battery powers, initiated a data upload request and when TCP connection is established between client and server then at the run time according to battery power and other parameters it is decided 25% data is uploaded at the server and keep the client live. We repeat the same experiment with same storage space and different battery powers and obtain the results as shown in the graph. Our proposed scenario cares about the consumption of battery power to keep the client live while existing system does not care about it and transfer fix data at every battery powers. In proposed mechanism if mobile node has more battery

power then data transfer rate is higher and if battery power is less then data transfer is also less and show an efficient use of resources.

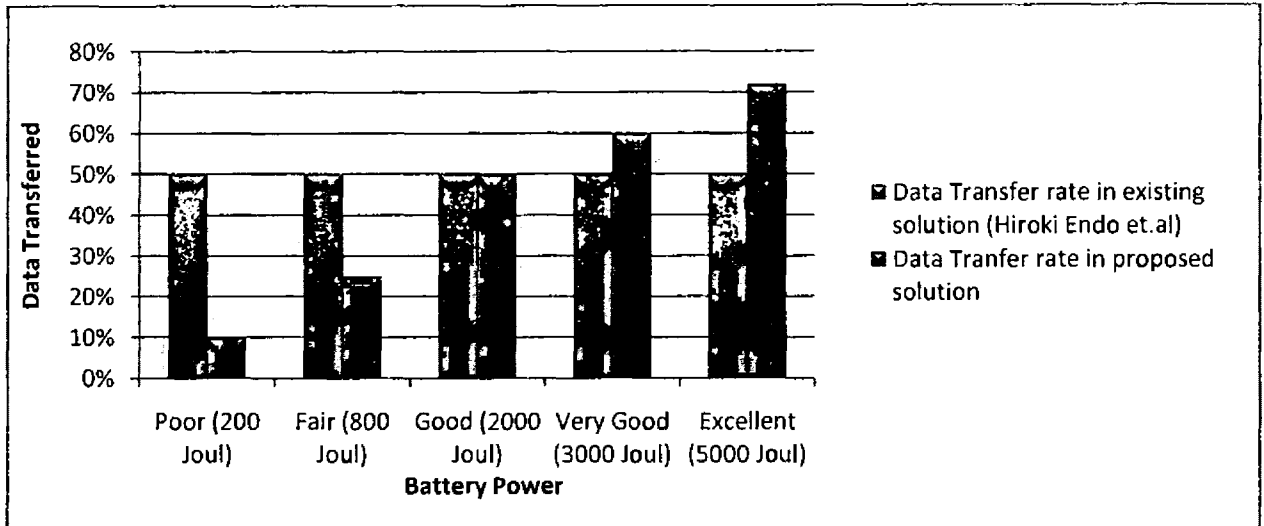


Figure 5.1: Data Transfer Rate vs. Different Battery Lives (sufficient Storage Space)

From the figure 5.1 it can be obviously seen that data transfer rate against different battery power among sufficient storage space. As battery power is increased data transfer rate is also improved among that storage space. For example if a mobile node has sufficient storage space and save 500 contact numbers in encrypted form in the local database then traversing procedure take some nanoseconds to traverse the required number and obviously consume a battery power, so the ratio of data transfer of mobile node vary with battery power. If one mobile node has more battery power it can transfer more data with same storage space.

## 5.2 Scenario 2: Data Transfer Rate vs. Battery Lives (Less Storage)

Same 5 clients with the battery power 200 Joules Poor, 800 Joules Fair, 2000 Joules Good,

3000 Joules Very Good and 5000 Joules Excellent respectively are taken like the previous scenario. The only difference we have made is decreased the memory from 500 Mb to 50 Mb. When mobile node has less storage space then consumption of battery power is less because less processing power is involved for traversal so battery power is saved during traversing data and transfer rate is higher as compared to those mobile nodes that have greater storage space.

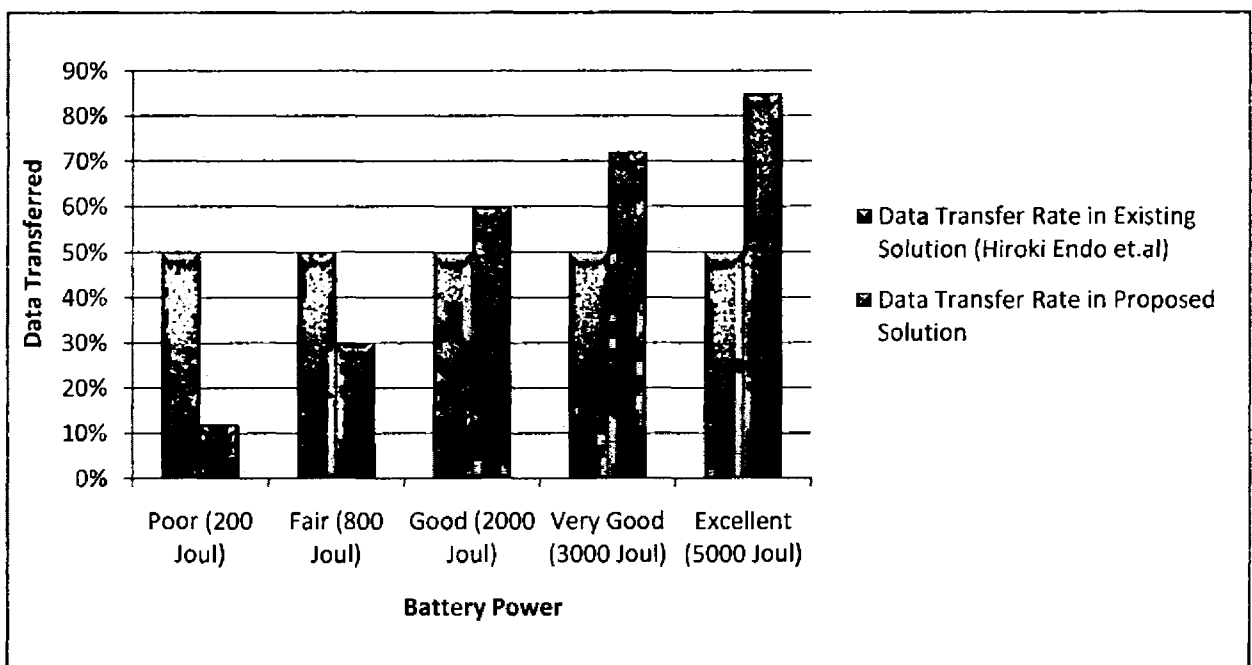


Figure 5.2: Data Transfer Rate vs. Different Battery Lives (Less Storage Space)

In this experiment we have considered data transferred rate of mobile node with less storage space among different battery powers. It can be clearly seen from the graph 2 data transfer rate with less storage space of mobile node is higher as compared to that one mobile node that have more storage space among varying battery powers. For example one mobile node have 50 megabyte storage space and have stored 100 contact numbers so when we search any



contact number the traversing tree is short and consume less battery power, due to less storage space less data management load so less battery consumption is cause the higher data transfer rate. So graph shows different data transfer rate with different battery lives which ratio is quite greater as compared to figure 5.1.

### **Case 1: Number of Request Launched and Served by Server**

We have applied an authentication mechanism on the server so if any spoofed request comes on the server our server never entertain that one request. Only legitimate users request is handling because every user have its own quota on the server when request come from any mobile device. Our server firstly checks and authenticates the user if that one user authentic then facilitates it otherwise refused. Existing server lack proper authentication mechanism so it tries to entertain all launched request that's may be included spoofed request and server have risk of flooding attack. Whereas our proposed solution applies a appropriate authentication mechanism on the server where user request to the server for uploading data. Server received the request and check the current status of mobile device user as well as calculate the percentage of data which allow to the user for uploading and show a response to the user among upload id and percentage of calculated data so as to data is encrypted by encryption algorithm at mobile device, as server is free it pull the data by upload id, at that time user upload the data at server, through this mechanism only legitimate users can interact with server and non authentic users can't access the server so our server saves from flooding risk. Hence server can properly response to the legitimate users.

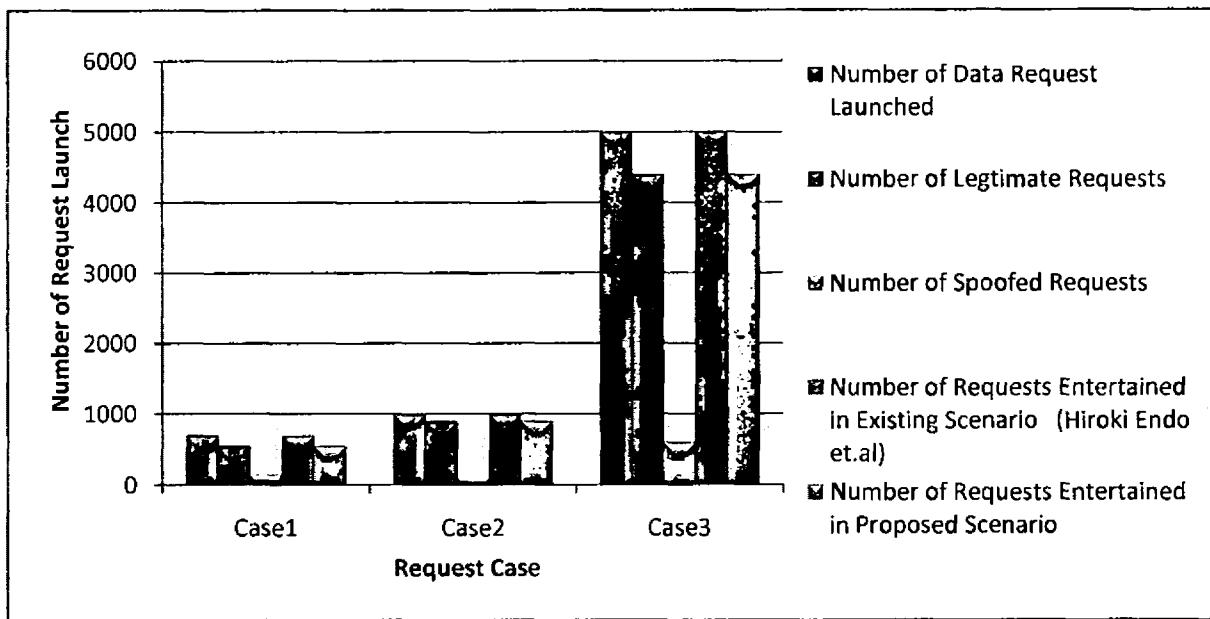


Figure 5.3: Number of Request Launched and Served by Server

Figure 5.3 shows when a specific number of requests are launched then almost 85% requests are legitimate requests and more or less 14% requests are spoofed requests. Existing scenario due to lack of proper authentication mechanism trying to entertain almost all launched requests included both spoofed and legitimate request and face a load on server. While in proposed scenario by the presence of authentication mechanism only legitimate request is entertained that is almost 85% of total launched requests. So load on server is decreased and our servers can properly response to the legitimate users.

### Case 2: Number of TCP Connection Initiated and Established

For sending the data transport layer protocol TCP is used. TCP connection is established which is always initiated with three way handshake and negotiate the actual connection. SYN packet initiates the TCP connection session then SYN/ACK packet moves to the client from

server side and finally ACK packet received the server from client side to acknowledge the session establishment. When TCP session establishment procedure is complete at this point able to start sending data. For handling the workload of server in proper manners sufficient bandwidth is necessary. Request from client side to the server may be timeout or rejected might be deferred response. When trying to establish all TCP connections including spoofed connection then in existing scenario server might face insufficient bandwidth or server busy in establishing connection receiving request and transmitting data cause connection fail. While in our proposed scenario we applied a proper safety measures on server when any request comes to the server it authenticate it if it is legitimate request then facilitated by our server otherwise refused and when our server facilitate one request other is in queue and when server is free then facilitate second third and so on . Hence our server save from flooding risk and insufficient bandwidth problem so all legitimate TCP initiated connection are established while in existing scenario lack of these safety measures a lot of connections fail.

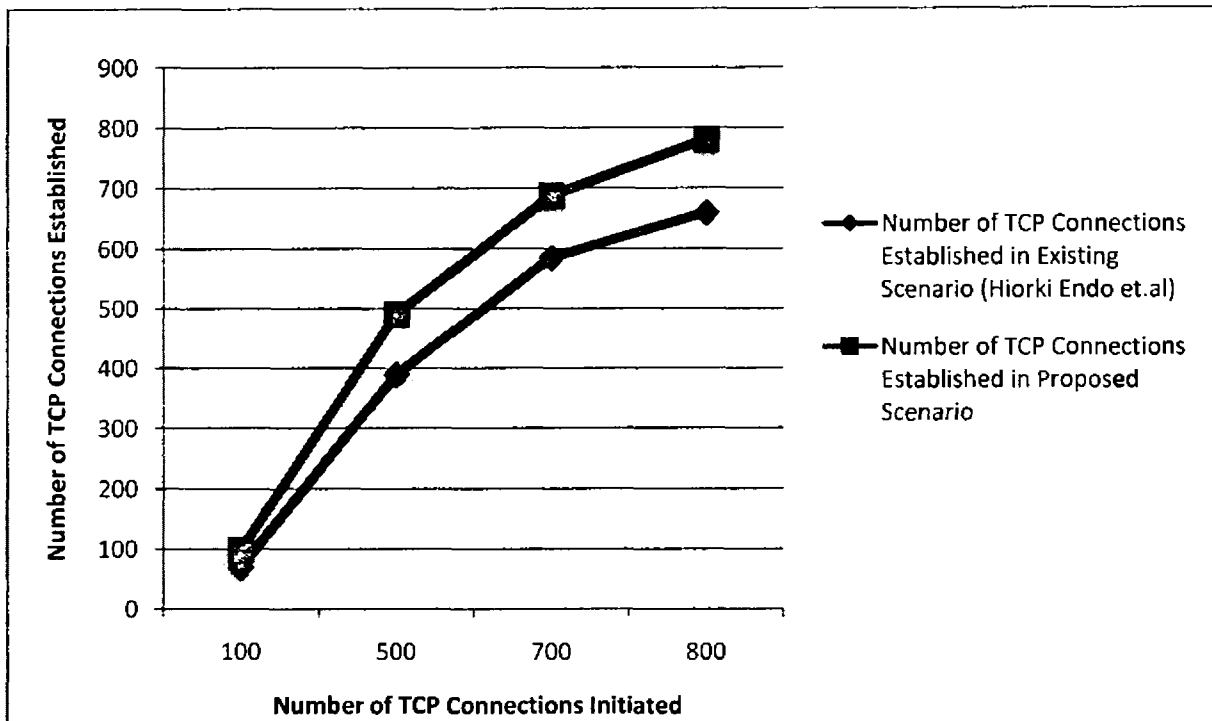


Figure 5.4: Number of TCP Connection Initiated and Established

Figure 5.4 shows TCP connection status how many connections are initiated and how many TCP connections were established. Graph 4 clearly shows the TCP connection status. By the existing scenario 88% connection are entertained from the initiated connection while our proposed scenario entertained the 97% TCP connection from the initiated connection of TCP. Our server entertains almost all legitimate initiated TCP connection.

### Case 3: Load on Server

We have applied an authentication mechanism on the server as well as on client side. And spoofed request is rejected or spoofed users can't access the server, only authentic and legitimate request access the server when server authenticates them then these legitimate

request by the server. So load on server is decreased and also traffic load is reducing in proposed scenario. While in existing scenario there is no authentication mechanism on server so server can't distinguish spoofed and legitimate request and try to facilitate all launched request that's why lots of load on server and threats of attack in existing scenario as compared to proposed scenario.

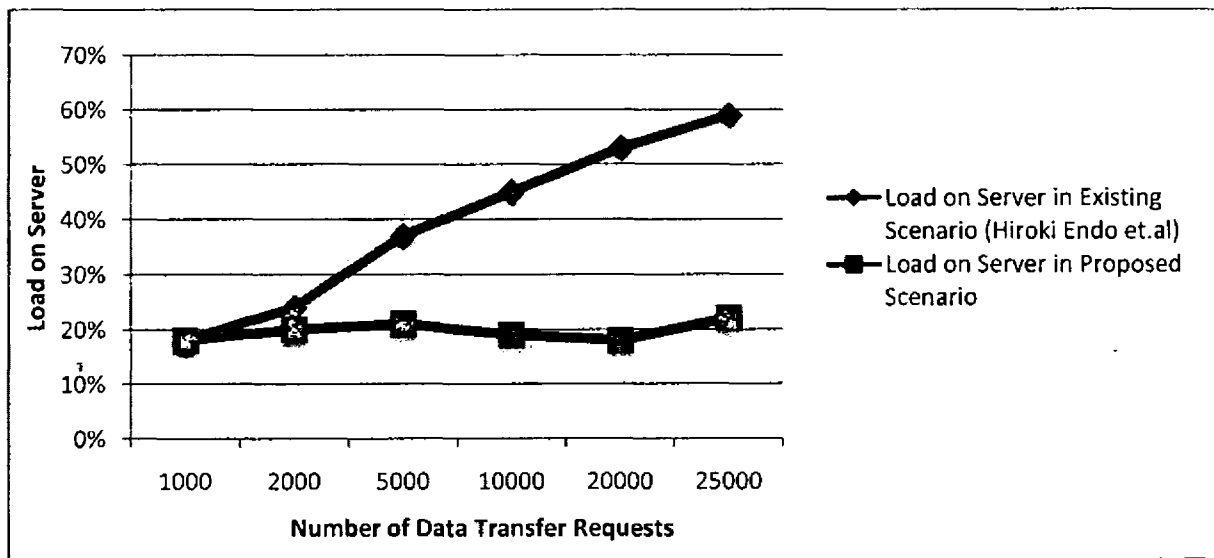


Figure 5.5: Load on Server

As graph shows the load of the server on how much data transfer request load. Load on server can be clearly seen from the graph 5. When a specific number of data transfer request is launched that holds both types of request genuine and spoofed request. By the absence of security measures and authentication mechanism in existing scenario there is 39% load on server, by the presence of these security measures there's load gets reduced by 20% load on server in proposed scenario.

## 5.6 Summary

We have thoroughly negotiated and analyzed the performance about data sharing technique based on simulation. Various parameters test and evaluate the technique and the performance and assessment of this technique has been shown by creating a graph.

# **Chapter 6**

## **Conclusion and Future Work**

## 6. Conclusions and Future Work

Now we have come to conclude our dissertation after a briefly portrayal on requirement analysis, proposed solution and methodology, implementation and results.

### 6.1 Conclusion

Telecom has always been part of our life style. But with the advancement there have always been the loopholes / vulnerabilities. Security is one of major concerns for telecom industry. Mobile and remote telecom devices while interact with each other are pretty much vulnerable because of lack of security. In absence of secure communication the data transferred over any medium give open handed to the hackers, also without security the spoofed request least to DoS (Denial of Service). In our dissertation our focal point is improving the effectiveness, performance, trustworthiness of data sharing on a network without compromising security. In favor of this purpose to mitigate the security threat for mobile devices during transferring data to the server we set and defined our research goal and we surpass to achieve our objective by designing the data sharing technique between sever and mobile. We in this thesis introduce a mechanism or technique that provides a secure transmission of data from mobile devices to the server so that no body / hacker can get any information from the hacked data or he can not alter any kind of data being transmitted. . When particular number of requests are launched then almost Average 85% requests are legitimate request out of total launched request and more or less 14% requests are spoofed request, by applying a proper authentication mechanism on server & client, it serve only legitimate request and reject the spoofed requests. that's why it establishes almost all TCP connection from the initiated TCP connection while existing scenario due to lack of authentication mechanism trying to entertain all TCP connection including



illegitimate connection and face insufficient bandwidth problem ,load on server and flooding risk attack. Whereas in proposed scenario 20% load get reduced on server as compared to the existing scenario, our server can properly response to the legitimate users and also our Server will save from flooding risk due to proper authentication mechanism on the server. This mechanism helpful to minimize the flooding attacks from illegitimate user and also security threats on the plaintext and reduces the load on server and entertains almost all initiated TCP connections. Our research work contains the simulated results which we assessed by using c# and MSSQL Server 2005 DB to pull off these targets. The evaluated results show the security measures during data transferring and also provide the security precautions during TCP connection establishing. And also provide the assessment of reducing load on server as compare to existing scenario.

## 6.2 Future Work

In this discussion we have focused some problems associated to security of data sharing and also we have solved these problems by providing a secure mechanism for data sharing. Innovative ideas are always welcomed in any area. For the endurance of current system atmosphere the enhancement is necessary for the reason that day by day technique and technologies are budding or for keeping alive these systems improvements and update is necessary. This technique can be more refined as discussed below.

In our research work mobile device encrypt the data or applied some security measures on the data and then transfer to the server, if we change the model of mobile device then definitely security measures type or encryption scheme change on that mobile model as compared to previous one model so how user get data from server in a compatible mode of this mobile model. Hence there is need a function at server side that provide mobile model compatibility

for converting the data from one form to another according to mobile model security measures or encryption scheme. That provides more ease to the users while we have no such function at server side this can be handle as a future work.

## References

- [1] K.Grenan, M.Storer, E.L.Miller, C.Maitzahn, "POTHARD: Storing Data for the Long-term Without Encryption" IEEE Computer Society Washington, DC, USA, 2005.
- [2] B.Bhargava, A.Burak Can, "Trust and anonymity in peer-to-peer system" Purdue University West Lafayette, IN, USA, 2007.
- [3]D.M.Blough, A.Subbiah, "Efficient proactive security for sensitive data storage" Georgia Institute of Technology Atlanta, GA, USA, 2007.
- [4]V.Kher, Y.Kim, "Securing Distributed storage: Challenges, Techniques, and System" Fairfax, Virginia,USA. StorageSS'05 November 11, 2005.
- [5]R.Hassan, Z.Anwar, W.Yurcik, L.Brumbaugh, R.Campbell, "A Survey of Peer-to-Peer Storage Techniques for Distributed File Systems" Proceedings of the International Conference on Information Technology: coding and Computing (ITCC'05), 2005.
- [6] M.W.Storer, K.M.Greenan, E.L.Miller, K.Voruganti, "POTSHARDS: secure long-term storage without encryption" USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference, 2007.
- [7] M.W.Storer, K.M.Greenan, E.L.Miller, K.Voruganti, "Pergamum: replacing tape with energy efficient, reliable, disk-based archival" Proceeding of the 6<sup>th</sup> USENIX Conference on File and Storage Technologies, 2008.
- [8] H.Kwon, S.Koh, J.Nah, J.Jang, "The Secure Routing Mechanism for DHT-based Overlay Network" Electronics and Telecommunication Research Institute (ETRI), 2008.10th International Conference on Advanced Communication Technology, 2008. ICACT 2008.
- [9] M.W.Storer, K.Greenan, E.L.Miller "Long-Term Threats to Secure Archives" Storage System Research Center, October 30, 2006, Alexandria, Virginia, USA, 2006.
- [10] H.Endo, Y.Kawahara, T.Asami "A Self-Encryption Based Private Storage System over P2P Distributed File Sharing Infrastructure" The University of Tokyo Graduate School of

Information Science and Technology, 2008. Innovations in NGN: Future Network and Services 2008. K-INGN 2008. First ITU-T Kaleidoscope Academic Conference

[11] N.S. Bathaee, M.R. Pakravan "A Bandwidth-Efficient Scheme for Distributed Storage System" Advanced Networks and Telecommunication System, 2008. 2<sup>nd</sup> International Symposium on **Issue Date:** 15-17 Dec. 2008

[12] Alexandros G. Dimakis, P.B. Godfrey, Martin J. Wainwright and K. Ramchandran "Network Coding for Distributed Storage Systems" Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94704. IEEE INFOCOM 2007 proceeding

[13] M. Ylianttila, E. Harjula, T. Koskela, J. Sauvola, MediaTeam Oulu Group "Analytical Model for Mobile P2P Data Management Systems" Department of Electrical and Information Engineering, Erkki Koiso-Kanttilankatu 3, FIN-90014 University of Oulu, Finland. 5<sup>th</sup> IEEE Consumer Communications and networking conference, 2008.

[14] M.A. Serhani, A. Benharref, R. Dossuli and R. Mizouni "Towards an Efficient Framework for Designing, Developing, and Using Secure Mobile Applications" World Academy of Science, Engineering and Technology 52 2009

[15] G. Ding and B. Bhargava "Peer-to-peer File-sharing over Mobile Ad hoc Networks" 2<sup>nd</sup> IEEE Annual Conference on Pervasive Computing ..., 2004 - cs.purdue.edu Department of Computer Sciences Purdue University West Lafayette, IN 47907, USA

[16] "Secure hash algorithm" <http://en.wikipedia.org/wiki/SHA-1>

[17] "Secure hash algorithm" <http://en.wikipedia.org/wiki/SHA-2>

[18] "Secure hash algorithm" <http://en.wikipedia.org/wiki/SHA-3>

[19] W. Stallings "Cryptography and Network Security" principles and practices, fourth edition

- [20] B. Schneier "Applied cryptography" protocols, algorithms, and source code in C , Second Edition, John Wiley & Sons, inc.
- [21] A. Klemm, C. Lindemann, and Oliver P. Waldhorst "A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks" - IEEE VEHICULAR TECHNOLOGY CONFERENCE, 2003.
- [22] Chung-Ming Huang, Member, IEEE, Tz-Heng Hsu, and Ming-Fa Hsu "Network-Aware P2P File Sharing over the Wireless Mobile Networks" IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 25, NO. 1, JANUARY 2007. The research is partially supported by the National Science Council of the Republic of China under the grant 95-2219-E-006-008.
- [23] Raphael C.-W. Phan, Member, IEEE, and M.Umar Siddiqi "A Framework for Describing Block Cipher Cryptanalysis" IEEE Transactions On Computers, VOL. 55, NO. 11, NOVEMBER 2006
- [24] Lee, D. Ohio State Univ., Ohio "**Hash Function Vulnerability Index and Hash Chain Attacks**" 3rd IEEE Workshop on Secure Network Protocols, 2007.NPsec2007.
- [25] Yi Lu Weichao Wang Bhargava, B. Dongyan Xu , Microsoft Corp., Redmond, WA "**Trust-based privacy preservation for peer-to-peer data sharing**" IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2006.
- [26] Fulu Li, David P. Reed, A. Lippman "Collaborative Storage with Mobile Devices in Wireless Networks for P2P Media Sharing" Wireless Telecommunications Symposium, 2008. WTS 2008. MIT, Cambridge, MA USA 02139.

[27] J.Dong, P. Liu, Z.Yuan “Distributed Storage and Retrieve of Similar Images in P2P System” ISISE Proceedings of the 2008 International Symposium on Information Science and Engineering-Volume 01. IEEE Computer Society Washington, DC, USA.

[28] Y.Ye,I-Ling Yen, L.Xiao, B. Thuraisingham “Secure, Highly Available, and High Performance Peer-to-Peer Storage System ” HASE Proceeding of the 2008, 11<sup>th</sup> IEEE High Assurance Systems Engineering Symposium. IEEE Computer Society Washington, DC, USA.