CHAPTER 5
PRIVACY PRESERVATION FOR FILE SHARING
SCHEME USING SECURED FILE BLOCK ID
WITH BINARY TREES

5.1 INTRODUCTION

Privacy is the protection of files or data from illegitimate disclosure and it is a widespread status of computer system proposal. Privacy is regularly more inadequate on the Internet and however more important while each user acts as both the customer and the originator. The obligation of privacy is mainly practical for peer-to-peer data sharing applications. In the previous years, status of systems for shared work and file sharing was enhanced significantly. The requirement for competent information distribution within the set of documents in the privacy preservation framework processed supplementary to attempt data management systems, in adding up to joint platforms for P2P networks. The research has been motivated by the great success of P2P networks from the work of Tomas et al (2010), the work progress is enhanced for the design and implementation of secured file sharing through online is presented by assigning a secured file block ID and a participant ID that provides users with explicit, configurable control over their file.

The massive volumes of confidential data are frequently composed and examined by applications using data mining. Such data contain shopping habits, illegal records, medical account and acclaim records, in the middle of
others. On the other hand, such records is a necessary benefit to business organisations and governments evenly to choose building processes and to present shared benefits, such as medicinal examination, crime diminution, national security. Otherwise, investigative such data discharge new threats to isolation and independence of the entity if not accomplished correctly.

The vulnerability to privacy, develops into authentic because data mining techniques are able to get hold of seriously accessible information from indefinite information that is not yet standard to database holders. To achieve privacy in a dynamic environment, it is essential to set up a protected relationship between peer nodes in the network. To get improved privacy preservation scheme, it is essential to manage with the reliability of the peer based on its performance in the communication environment. The research is motivated by the work of Huq et al (2008), for building file-sharing for advanced collaborating environment, in this study, binary tree representation is presented to enhance the privacy preservation file sharing mechanisms.

The level of the privacy is also being measured by sharing and distributing the policies of a peer. Owing to security and network overload, rising amount of collaborative data sharing needs to be addressed. As the quantity and combination of data enclosing user specific information raises, thrashing the requesters of data guides to research problems in privacy preservation scheme. At a procedural level, the privacy is easy to attain with central solutions.

If the user data is accumulated on a server in a data center, user guidelines about transmission can be merely imposed and information about user interests can be vigilantly limited or delayed on user request. However, the legitimacy is quite varied in practice. A lot of popular web services require users to spot away their division and manage rights as a status of service; sites regularly get hold of benefit of this to gather, build up and deal
out huge amounts of individual information regarding their users. Approximately each person on the Internet acts as a fulfilled producer and a contented consumer, with a mixed set of constraint on accessing the users’ privacy data. The work proposed by Jiang and Clifton (2005) was a measure to preserve the privacy of users distributed where it learns nothing that violates k-anonymity between the sites.

Privacy, the security of information sharing from unauthorized access is steadily essential on the Internet and yet increasingly more significant while every user acts as both the consumer and the producer. The lack of privacy is mostly engaged for peer-to-peer file sharing applications, in which users in the network shared file with each other and their actions are simply monitored by the unauthorised users. Several techniques have been obtainable to monitor the unauthorised access of files in the network, a work as proposed by Kantarcioglu and Clifton (2003) used certain privacy preserving tools (Secure Sum and Secure Set Union) to preserve the site information. The previous work described the secured file sharing using cryptographic key value pairs which shares the file in the middle of the users based on the key location of the file. But it does not offer a competent privacy preservation scheme for file sharing concepts. In this work, the secured file sharing concepts is implemented without disclosing users’ private data. Using binary trees, an assignment of file block id is allocated for each file to be shared and the participants involved in the file sharing mechanisms are assigned with a relevant id.

One could propose systems for every procedural model, e.g., one for anonymous publication as presented by Yang et al (2005), an additional for unidentified download, yet one more for prohibited sharing. A standard of the privacy scheme is to maintain a compilation of data sharing conditions competently within a distinct framework. In this work we present a Secured
file sharing concepts without disclosing users’ privacy data. Using binary
trees, and the task of file block id is done for every file to be shared and the
participants concerned in the file sharing mechanisms are assigned with a
relevant id.

5.2 BINARY TREE BASED FILE SHARING

This proposed work is also motivated by the great success of
applying binary tree for file-sharing in a secured manner and derived from the
work of Jonathan (2003), the work extend it to include file block id. The
Binary Tree (BT) is a tree data structures in which each node have at most
two child nodes, usually distinguished as “left” and “right”. Nodes with
children are parent nodes, and child nodes may contain references to their
parents. Outside the tree, there is often a reference to the “root” node (the
ancestor of all nodes), if it exists. Any node in the data structure can be
reached by starting at root node and repeatedly following references to either
the left or right child. A tree which does not have any node other than root
node is called a null tree. In a binary tree a degree of every node is maximum
two. A tree with n nodes has exactly n−1 branches or degree.

A Binary Tree is node-based data structure with satisfies the three
main properties:

i. The number \( n \) of nodes in a perfect binary tree can be found
   using this formula: \( n = 2^{h+1} - 1 \) where \( n \) is the depth of the tree.

ii. The number \( n \) of nodes in a binary tree of height \( h \) is at least
    \( n = h + 1 \) and at most \( n = 2^{h+1} - 1 \) where \( h \) is the depth of the tree.

iii. The number \( L \) of leaf nodes in a perfect binary tree can be
    found using this formula: \( L = 2^h \) where \( h \) is the depth of the tree.
iv. The number \( n \) of nodes in a perfect binary tree can also be found using this formula: 
\[ n = 2^L - 1 \]
where \( L \) is the number of leaf nodes in the tree.

v. The number of null links (absent children of nodes) in a complete binary tree of \( n \) nodes is \( n + 1 \).

vi. The number \( n-L \) of internal nodes (non-leaf nodes) in a Complete Binary Tree of \( n \) nodes is \( \lfloor n/2 \rfloor \).

vii. For any non-empty binary tree with \( n_0 \) leaf nodes and \( n_2 \) nodes of degree 2, \( n_0 = n_2 + 1 \)

where

\[ n = \text{the total number of nodes} \]

\[ B = \text{number of branches} \]

\( n_0, n_1, n_2 \) symbolize the number of nodes with no children, a single child, and two children respectively.

\[ B = n - 1 \text{ since all nodes except the root node come from a single branch} \]

\[ B = n_1 + 2*n_2 \]

\[ n = n_1 + 2*n_2 + 1 \]

\[ n = n_0 + n_1 + n_2 \]

\[ n_1 + 2*n_2 + 1 = n_0 + n_1 + n_2 \implies n_0 = n_2 + 1 \]

Because of that property, a BT is used inherently when it is necessary to keep the elements sorted, when a new element is inserted or when one is removed from the structure. This structure is used to implement
sets, multi sets and maps. In a set, there are no value associated with a key and a key can be present only once in the set. Moreover, not all the structures that are compared are real BT some of them are not even trees, but all of them will be implemented in terms of a set.

Each execution will hold up insertion, removal and search of value of an arbitrary type. Each tree can support value of any type; T referring to the type of value contained in a BT. The value is distorted to a key using a hash function.

There are different types of binary tree. The classification of different types of tree are collected from wikipedia and the work as proposed by Mehrzad et al (2011). They are

(i) A deep-rooted binary tree is a tree with a root node in which each node has at most two children.

(ii) An occupied binary tree occasionally called as a proper binary tree or firmly binary tree) is a tree in which every node other than the leaves has two children. Perhaps more clearly, every node in a binary tree has precisely 0 or 2 children. Now and then a full tree is ambiguously defined as a perfect tree.

(iii) An ideal binary tree is a complete binary tree in which all leaves are at the same depth or same level, and in which each parent has two children. This is obscurely also called a complete binary tree. An instance of a perfect binary tree is the chart of a person to a given depth, as every person has accurately two biological parents (one mother and one father). It is to be noted that this reverses the usual parent/child tree convention, and these trees go in the opposite direction from root at bottom.
(iv) A total binary tree is a binary tree in which each level apart from possibly the last is entirely filled, and all nodes are as far missing as possible.

(v) An unlimited total binary tree is a tree with a countably infinite number of levels; in which each node have two children, so that there are \(2^d\) nodes at level \(d\). The set of all nodes is countable infinite, but the set of all countless paths from the root is uncountable.

(vi) Objective binary tree is usually distinct as a binary tree in which the depth of the two sub trees of each node change by 1 or less, even though in universal it is a binary tree someplace no leaf is much farther away from the root than any additional leaf. Binary trees that are impartial according to this clarification have an unsurprising depth. This depth is equal to the integer part of \(\log_2(n)\) where \(n\) is the number of nodes on the balanced tree.

(vii) A disintegrate tree is a tree wherever for each parent node, presents only a single associated child node. This means that in a performance measurement, the tree will perform like a linked list data structure.

There is a diversity of dissimilar operations that can be performed on binary trees. Some are mutator operations, while others merely return positive information about the tree. The model is based on the work of Jonathan (2003), some of the operations are discussed for determining block file id.

- Insertion is a type of operation in which the nodes are able to be inserted into binary trees in between two other nodes or
additional following an external node. In binary trees, a node that is inserted is specific as to which child it contains

- Deletion is a mode of operation in which the Deletion is done by detached from the tree. Simply certain nodes in a binary tree can be unconcerned definitely

- Iteration - Frequently, one needs to visit every other nodes in a tree and observe the value, a procedure called iteration or enumeration. There are numerous universal orders in which the nodes can be visited, and each has valuable properties that are oppressed in algorithms based on binary trees. These are the three types of iterations
  - In-order: Left child, Root, Right child.
  - Pre-Order: Root, left child, Right child
  - Post-Order: Left child, Right child, Root

The binary tree file depiction method divides the file domain by means of the BGP. BGP approach partitions the preferred file repeatedly by straight lines in a hierarchical manner. First the stage is selected based on the suitable standard, to divider the entire file into the sub files. Using the similar measure the two levels are chosen to divide the two sub files resultant from the first partitioning. This process is repetitive till the termination principle is reached. The result of the recursive partitioning is the set of records that are referred to as a block of the segmented file.

The binary tree is also referred to as a Binary Gap Partition Tree depiction of the file. The non leaf node of the BGP tree is associated with the level and the leaf represents the block of the file. Each region of the file in the BGP tree representation has one or more attributes that describes the
characteristics of the file. The most critical aspect of the BGP tree construction process is the method used for selecting the partitioning level. At every stage of the repeated partitioning it includes the first stage in partition of the whole file.

BGP procession selection method consists of the major steps. They are described below:

(i) As the number of levels that bypass through the file under deliberation is countless and one have to quantise the space of all possible levels that partition the files. The level-quantization generates the finite set of levels that one requires to consider. Additionally, the levels in the files are represented using the level-quantization process. In common, the autonomous of the meticulous stage representation used, one of the two lines parameters is referred to as the level orientation.

(ii) Starting from the finite set of quantized levels, one has to choose the partitioning level of the file below deliberation. For this step one requests to describe the level collection criterion that can be used for selecting the partitioning the level.

As declared above, each step of the recursive partitioning of the BGP tree method has to recognize the limited amount of levels that require to be measured.

**Mathematical Model**

Let us presume that, present a N level of the binary tree, then the number of peak points used to embed messages with 2N. Once the pixel
difference $c_i$ satisfies $c_i < 2^N$ then if the message bit to be embedded is “0” the left child of the node $c_i$ is visited. Superior payloads necessitate the higher level. Conversely, the complete beneficiary wants to divide with the sender is the tree level $N$, because we propose an auxiliary binary tree that pre-determines multiple peak points used to embed messages.

![Figure 5.1 Auxiliary binary 2 level trees](image)

The steps to perform the binary tree operation:

- Resolve the level $N$ of binary tree
- Shift the data points from both sides by $2^N$ units. This is done to manage the over flow and under flow of data from the stack.
- Scan the entire data in inverse order and calculate the level differences between $y_{i-1}$ and $y_i$
- If $c_i \geq 2^N$ shift by $2^N$ units
To enhance the work progress, a design and implementation of secured file sharing through online is illustrated by assigning a secured file block and a participant ID that provides users with explicit, configurable control over their file. A FSP is developed to maintain a collection of users’ file assigned with its respective file and block ID without disclosing the users’ privacy data to the public. Then the file sharing is done with file block ID relating to the participant ID using binary trees which represents the exact location of data present in the file to be shared.

The binary trees keep all those files to be shared with a relevant file and block ID for each users’ file in a form of tree pattern framework. The proposed secured file sharing using B-tree is optimized for systems that read and write large blocks of files in a chronological manner. A tentative evaluation is done with several user clients in terms of communication key round, number of participants and the size of the file for exchange to estimate the performance of the proposed privacy of file sharing using secured file block ID using with binary trees [PFSBT].

It also maintains information of file about file ID, block ID, owner ID of the respective file and the shared participant ID for that shared file. The file security packet maintained valid information about the user and the participant and none other user can access or modify the data maintained by FSP, since it allowed an access to the file only the participants who have a valid ID or valid member of the file security packet mechanisms. With the binary trees, we can easily identify the location of file and the user who holds the respective file.
5.3 FRAMEWORK OF PROPOSED PRIVACY PRESERVATION OF FILE SHARING USING SECURED FILE BLOCK ID WITH BINARY TREES

The proposed work is efficiently designed for providing a secured file transferring scheme among the networks in the system by assigning a secured file block ID to each file maintained by the user and associated with each participant using binary trees. The proposed privacy of file sharing using secured file block ID using with binary trees [PFSBT] is operated under two phases.

The first phase assigns a secured file block ID using file security packet. The second operation is to provide a privacy preservation scheme of file sharing approach using binary trees. The set of participants namely the P1, P2, P3, P4, P5 are used as a set of participants to share the file securely with the binary tree usage. The binary tree forms a secure tree structure for file sharing.

The architecture diagram of the proposed privacy of file sharing based on secured file block ID using binary trees [PFSBT] is shown in Figure 5.2.
Figure 5.2 Architecture diagram of the proposed PFSBT

The first phase describes the process of evaluating the secured file block ID scheme for every file sustained by the users using a file security packet. A FSP maintains a security relevant data for files in the form of FSP structure. Using FSP, a file block ID is assigned and processed for file sharing mechanisms. The FSP data can be examined by users other than the security product. The FSP is mapped by file and block ID, so others should not use this
mapping to create or directly modify the FSP and should not make their own security or audit decisions based on the contents of the FSP.

The second phase describes the process of privacy preservation scheme for file sharing approaches using a secured file and block ID for each users’ file with binary trees. The privacy preservation scheme of file sharing approaches is done with secured file and block ID relevant to each participant ID using binary trees.

The Figure 5.2 represents the entire process of the proposed privacy preservation scheme for file sharing approaches using binary trees by assigning a file and block ID for each file maintained by the users’ involved in the network. The assignment of each file ID is done with the help of file security packet scheme.

5.3.1 Turning over File ID and Block ID using FSP

A file security packet is a storage space system which assigns an ID for each file and the participant and none other user can modify or use the data in the FSP without the knowledge of the owner of the respective file. A participant in the network has one or more files which are ready to be shared with the other users in the network. For each participant, it is necessary to assign an ID by tracing the actions of that participant in the communication in which the concern user has involved before.

When the two peers attach, they substitute file list messages, File list messages are condensed XML counting attributes recounting the size, name, date shared and other meta-data for files, which a scrupulous participant has permissions. For each confidentially shared file the meta-data comprises a potential that is used as a symmetric encryption key for exercise during transfers.
Instead of sharing all data visibly with an active set of participants, file security packet clearly classify the trust level of a determined set of participants and files using their respective IDs (by default peers are untrusted). Second, instead of integrating information about which users have which files, put isolated data sources by torrenting object lookups during the overlay. Third, instead of sources transferring data openly to receivers, data transfers are done using participant and file block identities.

![Diagram of file security packet](image)

**Figure 5.3 Process of file security packet**

The Figure 5.3 clearly shows that a file security packet will maintain all the details about the participants and files. For a file, FSP comprises of FID (File ID), PID (Participant ID), Shared PID, BID (File Block ID). The FSP maintains the information of participant about the participant ID and number of files in which each participant has.

### 5.3.2 Representation of Binary Tree for Privacy Preservation Scheme

After assigning the file and participant ID or each participant and files which they maintained, the preceding step discuss about how the files are
shared in a secure manner without disclosing the private data of the participant using binary trees. For each participant involved in the communication will follow the binary tree pattern framework for each files they maintained. The other participant (User B) could share the file of participant (User A) using block ID than file ID. The file ID provides information about the location of file.

![Binary Tree Representation](image)

**Figure 5.4  Binary representation for privacy preservation scheme using FSP**

A block ID refers to the location of the exact data content in the file. Rather than sharing with the file ID, the file block ID file sharing consumes less time to achieve the file sharing concepts since it specified the exact location of the data content to be shared. The Figure 5.4 describes the binary tree representation for the secured file sharing scheme. A user (P1, P2, and P3) has one or many files (F1, F2, F3, F4, F5, and F6) to be shared.

For each file, a file ID and block ID are specified and they are represented as specified in Fig 5.4. A straight line in Figure represents the file ID and block ID a file have and a dotted line represent the sharing of the content of the file by another user. Using file ID and block ID, users can share
their files with the other users in the network. A file ID specifies the location of file in the file security packet and the block ID refers to the specified data location on the file security packet. So, it is easy for the user to choose the file block ID to share retrieve the content from the file in a secure way.

When sharing data with binary tree representation using FSP, disclosure is limited by familiar mechanisms: strong identities, capabilities and end-to-end encryption. So, the privacy of the data content in the file stored in a secure way and only the authorised user can access it in a reliable manner. The pseudo code below describes the performances of the proposed privacy of file sharing using secured file block ID using with binary trees [PFSBT]:

**Input:** Set of users U and Set of files F

**Step1:** Each user might have one or more F

**Step2:** Apply FSP // Assigning secured File Block ID

**Step3:** For each user U

**Step4:** Assign UID

**Step5:** Count the number of files user has (n)

**Step6:** End For

**Step7:** For each file F

**Step8:** Assign FID

**Step9:** Assign BID

**Step10:** Find the UID of the respective file F

**Step11:** End for
Step 12: FSP maintains all the above mentioned details in it i.e., it maintains only the authorised data // Privacy Preservation using Binary tree representation

Step 13: Form a binary tree structure based on U and F

Step 14: For each U (UID)

Step 15: There might be some F (FID)

Step 16: For each F, there might be some BID (Block ID)

Step 17: End For

Step 18: End for

Step 19: Allow the users to share their file based on BID using their own UID, since BID refers to the exact location of the data content of the respective file

Step 20: End (Privacy preservation achieves)

The above algorithm describes the process of the proposed privacy file sharing using secured file block ID with binary trees [PFSBT]. The first process is to assign a secured file block ID to the files and participants who are involved in the communication process. The FSP maintains the entire file and participant ID in it and acts as like a file allocation table. After assignment of secured file block ID, the binary tree representation is done by splitting the file and block ID for each file maintained by the respective participant. Through the tree framework, the files can be accessed in a secure manner without disclosing its privacy data.

5.4 PERFORMANCE EVALUATION

The proposed privacy of file sharing using secured file block ID with binary trees [PFSBT] is efficiently designed and used for providing a
secured way of file sharing approach using binary tree representation. The implementation of the proposed PFSBT was done with Intel P-IV machine with 2 GB memory and 3 GHz dual processor CPU. At first setup, the secured file block ID is assigned using file security packet scheme and the representation of binary trees are used to form a secured file sharing approach by using file block ID.

When compared to an existing cryptographic key value pairs and the group key management (Combined with ABE (Attribute Based Encryption) and traditional public key cryptography), the proposed PFSBT provides a secure file sharing mechanism in a reliable manner by implementing file security packet and binary tree representation. The binary tree is formed with the tree pattern framework by simply assigning the corresponding file and block ID to its respective participants. The performance of the proposed privacy of file sharing using secured file block ID with binary trees is measured in terms of:

(i) FSP efficiency
(ii) Tree Building time
(iii) File sharing time
(iv) Throughput

5.5 RESULTS AND DISCUSSION

The experiment is conducted with privacy preservation of file sharing scheme using the secure file block id. The secured File block id with the binary trees allows explicit mutual authentication in maintaining the privacy in sharing the files shows better resistance rate when compared with an existing Cryptographic Key Value Pairs (CKPV) model and ABE Model. An experimental evaluation is processed with several user clients in terms of
communication key round, number of participants and the files utilized for exchange to estimate the performance of the proposed PFSBT.

5.5.1 FSP Efficiency

The FSP efficiency describes the effectiveness of secure transformation of the files over the network the user has.

\[
\text{FSP efficiency} = \frac{\text{No. of files transferred securely}}{\text{Total no. of files}} \times 100 \quad (5.1)
\]

The Table 5.1 describes the file security packet efficiency for assigning a secured file block ID mechanism. The outcome of the proposed privacy of file sharing using secured file block ID with binary trees is compared with an existing CKVP and ABE with traditional cryptography model.

| No. of Participants | FSP Efficiency (%) |          |          |          |
|---------------------|--------------------|----------|----------|----------|
|                     | ABE Model          | CKVP Model| PFSBT Model|
| 5                   | 16                 | 10       | 24       |
| 10                  | 24                 | 15       | 48       |
| 15                  | 38                 | 22       | 62       |
| 20                  | 45                 | 34       | 75       |
| 25                  | 56                 | 37       | 82       |
| 30                  | 61                 | 45       | 87       |
| 35                  | 69                 | 49       | 93       |
The efficiency of the File Security Packet is also measured in terms of number of users involved in the session. As a result, an increase in users is supposed to considerably increase the FSP efficiency. The performance graph of the proposed Privacy of File Sharing using Secured file block ID with Binary Trees (PFSBT) for secure file sharing with an existing CKVP and ABE as shown in the Figure 5.5.

![Figure 5.5 Number of participants Vs. FSP efficiency](image)

The Figure 5.5 describes the effectiveness of file security packet when number of users in the communication increases. In the proposed PFSBT, the secured file sharing is done using a file security packet scheme which assigns a secured file block ID and file ID to all the files present. FSP allowed assigning id only to the authorized users and files. So, the security level of the FSP is high. The FSP efficiency is measured based on the security level of the files and the users it holds.

In an existing cryptographic value pairs, the security level is low since it does not use any security mechanism specifically to protect all the
files to be shared for a communication. Compared to an existing cryptographic key value pairs, the proposed privacy of file sharing using secured file block id with binary trees outperforms well by maintaining the file security packet and the variance is 30-40% high in the proposed PFSBT.

5.5.2 Tree Building Time

The tree building time is how long it will take time to build the binary tree structure framework for each participants and files used in the communication level.

| No. of Sequential Files | Tree Building Time (in Sec) |          |          |          |
|-------------------------|----------------------------|----------|----------|----------|
|                         | ABE Model                  | CKVP Model| PFSBT Model|
| 2                       | 3.4                        | 3.8      | 1.3      |
| 4                       | 5.4                        | 6.8      | 2.4      |
| 6                       | 9.3                        | 10.2     | 3.6      |
| 8                       | 10.1                       | 12.1     | 4.1      |
| 10                      | 10.7                       | 13.2     | 5.2      |
| 12                      | 11.5                       | 14.3     | 5.7      |
| 14                      | 13.1                       | 15.1     | 6.8      |

The Table 5.2 describes the time taken to build the binary tree for achieving the privacy preservation scheme. The outcome of the proposed privacy of file sharing using secured file block id with binary trees is compared with an existing CKVP and ABE Model.
The Figure 5.6 describes the binary tree building time based on the number of sequential files a user have. Since the proposed followed the representation of binary tree, the privacy preservation is highly achievable and it allows the participants for an easy access of the file contents the participants have, the binary tree is built based on participant ID and file block ID scheme. The tree building time is measured in terms of seconds.

When compared to existing methods, the proposed privacy of file sharing using secured file block id with binary trees, efficiently built the tree, with the file block evaluated from file security packet and the variance is 20-30% better in the proposed PFSBT.

5.5.3 File Sharing Time

The file sharing time is the time taken to share the respective file from the tree framework among all the participants in the communication level.
Table 5.3 Number of participants Vs. file sharing time

| No. of Participants | File Sharing Time (in Sec) | ABE Model | CKVP Model | PFSBT Model |
|---------------------|-----------------------------|-----------|------------|-------------|
| 5                   | 9.8                         | 10.2      | 8.2        |
| 10                  | 13.9                        | 15.4      | 11.1       |
| 15                  | 17.6                        | 19.2      | 13.3       |
| 20                  | 21.7                        | 24.8      | 18.9       |
| 25                  | 26.8                        | 30.0      | 20.5       |
| 30                  | 27.5                        | 31.5      | 22.6       |
| 35                  | 29.0                        | 33.2      | 23.9       |

The Table 5.3 describes the time taken to share the file in a short interval of time for achieving the privacy preservation scheme. The outcome of the proposed privacy of file sharing using secured file block ID with binary trees is compared with an existing CKVP and ABE Model.

Figure 5.7 Number of participants Vs. file sharing time
The Figure 5.7 describes how long it will take to share the file among the users involved in the communication based on the number of users present. In the proposed PFSBT, the file sharing is done with a secured file block ID. The binary tree has set of participants and each participant might have a set of files specified as file id and block ID. The file ID specified the location of file in the file security packet and the file block ID specified the exact location of the content of the file to be shared. The proposed PFSBT shared the file in the network using file block ID, so the consumption of time to share the file among the users will become less.

The file sharing time is measured in terms of seconds (secs). Compared to an existing cryptographic key value pairs, the proposed privacy of file sharing using secured file block ID with binary trees, efficiently share the file without disclosing its privacy data and the variance is 10-30% better in the proposed PFSBT.

5.5.4 Throughput

It refers to how much data transferred from one location to another in a given amount of time. It is used to measure the performance of hard drives and RAM, as well as Internet and network connections. It is measured in terms of percentage (\%).
The Table 5.4 describes the throughput for the successful file sharing for the participants through binary trees. The table compares the throughput of the proposed Privacy of File Sharing using Secured file block ID with Binary Trees (PFSBT) for secure file sharing with the existing CKVP and ABE Model.

The Figure 5.8 describes the throughput for successful sharing of the file using Privacy of File Sharing using Secured file block ID with Binary Trees (PFSBT). In the proposed PFSBT for secure file sharing, the number of participants indicates the number of users available in a specified session. The throughput obtained during the PFSBT is high when contrast to an existing CKVP and ABE Model.
Figure 5.8 Number of participants Vs. throughput

Basically, a Privacy of File Sharing using Secured file block ID with Binary Trees technique is based on sharing of the file for multi-path computational model with secure block id. As shown in Figure 5.8, the PFSBT performs better than CKVP model and ABE Model. The performance graph of the proposed Privacy of File Sharing using Secured file block ID with Binary Trees for throughput is shown in the Figure 5.8.

When compared to an existing models, the proposed Privacy of File Sharing using Secured file block ID with Binary Trees (PFSBT) for secure file sharing, in terms of throughput performs well. The variance would be approximately 20 – 30 % higher in throughput of the proposed Privacy of File Sharing using Secured file block ID with Binary Trees (PFSBT) technique.

Finally, it is observed that the proposed privacy of file sharing using secured file block id with binary tree efficiently achieved the privacy scheme by implementing a file security packet and the representation of binary tree based on the FSP and shared the respective file with the users.
based on file block id. The proposed PFSBT allowed the users to share the file in a short interval of time without releasing its privacy data.

5.6 SUMMARY

An incredible growth in internet made users to share everything through online. To share a file each other online, without disclosing the private data, this chapter efficiently described the process of sharing the file in a secure manner using file security packet mechanism with binary tree representation. To improve the secured sharing of file among the users’, the work presented a File security packet scheme. The FSP played an important part of the privacy file sharing mechanism, since it has maintained all the privacy data of the user and files. By using file security packet, the file sharing is being done reliably for many users using the representation of binary trees. The experimental evaluation presented here described the performance of the proposed PFSBT carried out with the number of participants and the files they hold for privacy file sharing mechanism. The results are shown that compared among existing CKVP and ABE Model, the proposed PFSBT provides an efficient secure file sharing mechanism in the network.