Black-Box: A New Secure Distributed Peer-to-Peer File Storage and Backup System

Syed S. Rizvi
Department of Information Sciences and Technology
Pennsylvania State University
Altoona – PA USA
srizvi@psu.edu

Abdul Razaque
Department of Electrical Engineering and Computer Science
Cleveland State University
Cleveland – OH USA
arazaque@csuohio.edu

***Abstract*—The amount of valuable data pertaining to users, such as digital documents, images, and various other important files, has grown exceedingly and needs to be backed up safely. The two most widely used existing backup solutions are the removable devices and the online storage. Problems can arise in the external device backups, for example it takes a long time to burn and copy data files which can result in frequent data loss. Online storage proved to be a better option, but when a large amount of data is to be stored, these third party services become costly. In this paper, a back-up file system is designed, which allows users to store files on their extra disk space. Black-Box is safe for users to back-up their important and secret data. Traditional redundancy schemes focus on storage capacity and computation. However, the proposed Black-Box approach is supported by redundancy mechanisms which include replication and erasure coding.**

***Index Terms*—Information security, Peer-to-peer system, Hierarchical codes, replication, encryption and decryption, cryptography**

I. Introduction

The main motivation behind the implementation of this project is the extremely high demand for user data backup. The amount of data stored on peer-to-peer networks by users has increased over the years. Data reliability presents several threats, including errors by users, data theft, etc. [1]. Ordinary users cannot afford data backup systems, as they are very costly, so there is a need for inexpensive file storage systems. Although we have many backup systems, such as BitTorrent Sync, Tahoe-LAFS, PAST, and Freenet, they do not serve as perfect file storage back-up systems. Therefore, in response to the defaults of previous systems, Black-Box is proposed as a secure, fault tolerant distributed peer-to-peer file system [2].

Black-Box is a peer-to-peer distributed back-up file system that includes applications of cryptographic schemes, which ensure security and connection reliability among clients. Black-Box mainly works as a client that does not need a centralized server but has access to the internet, which assures secure and uninterrupted connections.

978-1-5090-4300-2/16/$31.00 ©2016 IEEE

The Black-Box system achieves the following objectives: reliability, confidentiality, integrity and fairness. Users are able to back up their files that are good in quality and performance. Files stored in another computer cannot be easily retrieved back by the other users, ensuring system confidentiality [3]. Black-Box is a virtually free service and the resultant cost is equally shared among all users. The idea of peer-to-peer storage systems is to apply redundancy to the data and distribute it among all the peers in the network [4]. The addition of redundant data is done by trivial replication. The two levels of replication schemes used are whole file and fragment [5]. In addition, erasure coding, which is essentially file chunking and encryption/decryption, is performed on the required data. The effective use of bandwidth is also discussed in this peer-to-peer storage system [6].

The remainder of the paper is organized as follows: Section II presents problem identification and significance. Section III presents the literature study. Section IV presents our proposed Black-Box peer-to-peer file storage and backup scheme. In Section V, we present our analytical model to formalize AES. Section VI presents the results of the system. Finally, we conclude our paper in Section VII.

II. Problem Identification and Significance

In any peer-to-peer network, peers have access to join or leave the network at any time. Due to this irregular peer availability, assuring stored data availability is the most challenging problem. Redundancy mechanisms serve as a solution to the fore-mentioned problem. These can be achieved by using replication and erasure codes. Erasure coding is more effective when compared to replication because it provides higher data availability.

Replication is used when peer availability is high and when low erasure coding serves as a better option. Erasure coding also results in less bandwidth and storage when compared to replication. The main drawback of erasure coding occurs when the node fails. The naïve approach serves as a solution for this. In addition to the redundancy mechanism, policies are implemented to recover lost or temporarily unavailable data repair.

In this paper, we propose a productive model for a peer-to-peer file storage and backup system with numerous

functionalities such as integrity, acquaintance, and reliability services. With our model, recovery of the data backups will be easily available. This can be achieved by the users where the users will be authorized to retrieve their work when they are associated in the partnership networks.

Files that a user wants to back up will be stored in an efficient folder much like Dropbox's folder. All files in this folder will be encrypted and distributed to peers to save. In other words, every file is encrypted by the owner of the file before it is distributed to peers in the network. Each user must have their own AES (Advanced Encryption Standard) key stored in their database, and it is extremely important that the client should not free the key. Here we are using decentralized network architecture because individual nodes can easily go offline.

There remains some work to be done on this subject, so we have made quite a few assumptions in this function that is utility function. However, we are only considering that users had two decisions: the data quantity that they are storing into the system and the amount of storage capacity that they are offering. The work has to be done towards an extension of the model where users can also choose the proportion of time they are online.

III. Literature Study

This study focuses on Black-Box and products that are quietly accompanied with Black-Box, such as Freenet. Freenet is considered as a wireless peer-to-peer data transfer application and a platform for censorship resistant communication. It delivers information by way of a decentralized distributed data store to keep the data and transfer secure. It provides a suite of free programming for distributing and conveying on the web without trepidation of oversight.

Many of the researchers suggested that this tool can provide ambiguity on the Internet by providing storage for small encrypted data snippets distributed on the computers [7]. These users are connecting only through midway computers, which are simultaneously passing on the requests for content and sending them back without acknowledging the contents of the full file. It is similar to internet routers routing the packets without knowing anything about files (excluding Freenet has caching). This is considered as a layer of strong encryption with no reliance on centralized structures. This allows users to publish namelessly or recover different sorts of data. It acts as a trusted third party system by a trusted third party communication.

PAST is a system that is composed of nodes oriented to the Internet; in this, each node has the capability of initiating routing client requests to insert or help in retrieving the data files. PAST is considered a large-scale, persistent peer-to-peer storage utility resource. PAST also acts as a global storage utility that provides scalability, high availability, persistence, and security [9]. BitTorrent Sync tool acts as a client for syncing data between many authorized peers who share a secret key between them. In this technology, two nodes or user devices must be online in order to synchronize files between them. This sync tool encrypts data using an AES key, which is either default or user definable [10]. Tahoe LAFS is considered a distributed file system which stores files on multiple computers to protect against hardware. It is based on the principle of least authority file store. Like Freenet, it is also an online backup system.

IV. Proposed Black Box Scheme For Peer-to-Peer File Storage and Backup

The strategy of Black-Box is characterized by a series of exposed systems and is a circulated peer-to-peer system which maintains the packing of files and providing a backup. Backups guarantee security for the files which were packed, and the equality and data transmission consistency between the systems in the network. The Black-Box acts as a patron which has contact to the internet, but has no need of a consolidated server. The patron assures that even if a segment of buyers has a chance to misuse the system, the dependability will be maintained to secure the system without allowing the involvement of obligation objects into the system.

The main aspects involved in Black-Box are confidentiality, data maintenance, and data reliability; also, in terms of storage, Black-Box includes data redundancy mechanisms such as erasure coding and replication. When using these schemes only, storage efficiency is achieved. In order to repair large amounts of lost data, more communication bandwidth is consumed, which is expensive. Here we propose hierarchical codes for the effective usage of communication bandwidth. And also we introduce an adaptive repair scheme for reducing communication bandwidth from a different perspective.

A. Replication Process

Replication serves as a solution for data unavailability. Here the information is replicated n times and each replica is stored in different peers. The replication process is shown in Fig. 1.

Although data is lost in one peer, the replica of it will be available in another peer, thereby masking the data unavailability. If in any case the replica is lost, recovering or rebuilding is simple in replication, as it is easy to recreate an exact replica.

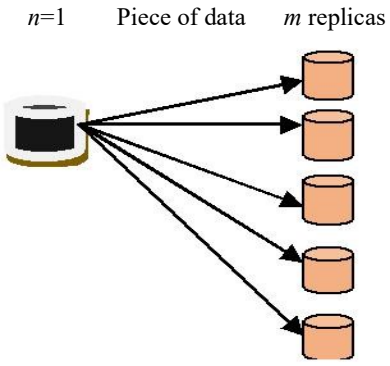

Fig.1. Replication process

B. Erasure Coding

The most effective redundancy scheme is erasure coding. In our equation below, the data object is divided into *m* fragments and they are recoded again to *n* fragments, where *n* must be always greater than *m*. The rate of encoding is given as follows:

$$r = \frac{m}{n} \; (r < 1) \qquad (1)$$

Fig. 2 shows erasure coding where *m* fragments are divided into *n* fragments.

C. Hierarchical Codes

Hierarchical codes are implemented in order to achieve storage efficiency, reduced communication bandwidth and computational complexity. Hierarchical codes make use of linear coding and fragment replication (the fragments are replicated and stored in different peers instead of a total file).

Here, *f1* and *f2* are the original fragments and *p1*, *p2* refers to parity fragments.

For the hierarchical (*k0*, *h0*) code graph, the parity fragments are obtained by a linear combination of original fragments. The code graph for (2,1) is given in Fig. 3. The parity fragments *p1*, *p2*, *p3* form a group *G* (2,1). This group is replicated *g1* times (here *g1*=2) to get groups *G* (2,1) and *G* (2,2). This group adds another *h1* parity fragments (here *h1*=1) to get *p7*. These parity fragments are obtained by a linear combination of original fragments *f1*, and *f2*. This gives the resultant hierarchical code graph (*d, k*), where *d*=4=*g*1*k*0 and *k*=3. Hence, we get (4, 3) code graph.

Hierarchical codes are developed in order to reduce the repair degree thereby reducing overall consumption of communication bandwidth. The hierarchical code graph (4, 3) is depicted in Fig. 3.

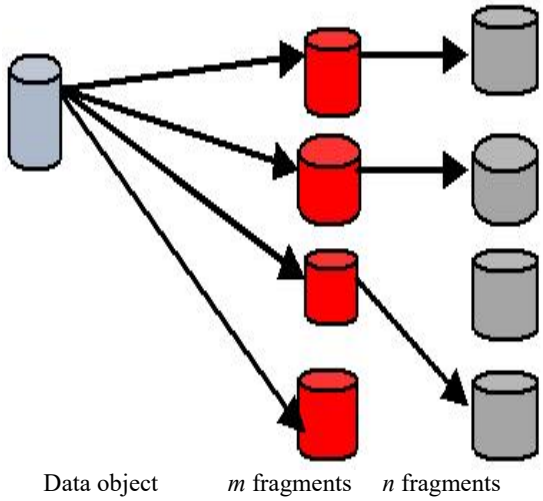

Fig. 2. Erasure coding

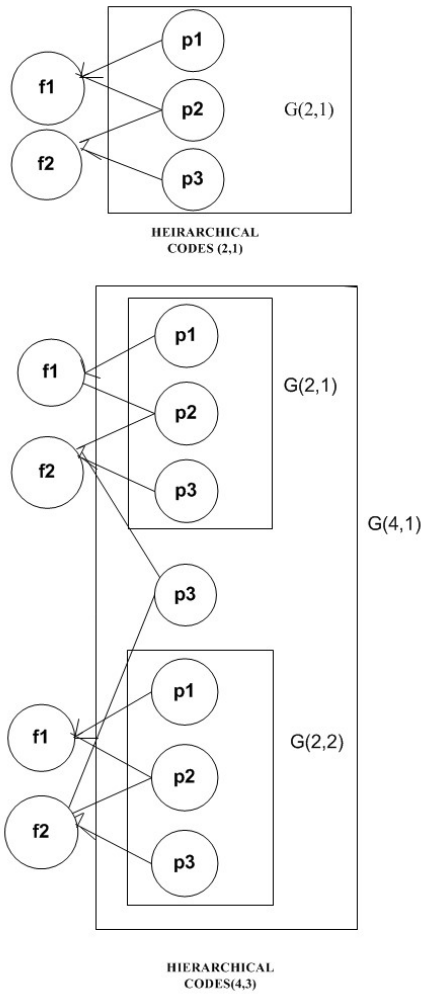

Fig. 3. Hierarchical codes

D. Adaptive Repair Scheme

The main objective of this repair policy is to reduce the quantity of data needed to be transferred for data repair. This adaptive repair scheme smooths the repair rate and can be proactive or reactive. Proactive schemes use a constant repair rate and reactive uses a variable repair rate.

In this scheme, an adaptive control scheme is developed that adopts the repair rate to change the system behavior. It consists of three components: system, estimator, and controller. System is defined as the status of peers to store a particular object. We have three parameters of the system: μ (disconnection rate), λ (reconnection rate) and *Pdeath* (death probability). The estimator estimates the parameters by observing the systems output. Finally, the controller receives inputs from the estimator and from online peers.

Algorithm 1 The basic operation for encryption and decryption using AES key for file storage and backup.

1. Initialization (F: File; C: Chunks; P :Password; D_E: Data Encrypted; D_d: Data Decrypted; E_k: Encrypted key, D_k: Decrypted key
2. Input (P)
3. Output (D_E, D_d)
4. Set AES for P
5. **if** D_k==P **then** Access to the key accepted
6. **else if** D_k != P **then** Access to the P denied
7. F= (P, D_k, E_k)
8. End if
9. End else if

E. File Encryption and Decryption Process

The design of a network is implemented in terms of confidentiality, integrity, consistency, equality, security, and storage of files by providing backup to the system by using the methodologies related to these features. In order to improve the security provided to the system, the Black-Box design is used in a way that the cryptographic methodologies are used to encrypt the data that is transferred. This is so the data cannot be altered by any other patron. Confidentiality is the essential term in the network design since the information in the file has to be maintained and transferred without any errors. Therefore, an encryption method is an important aspect we implement in this paper. The basic operation for encryption and decryption using AES key for file storage and backup is shown in Algorithm 1.

F. Costs Analysis of Different Types of Operations

In this section, we perform the cost analysis for different types of operations that our proposed scheme typically performs. These operations include the following: storage, communication, and computation. In the later part, we use this to determine the performance of our proposed scheme.

Storage: Stored files consume more storage space than the original file.

$$Storage = (m+n)_{\bullet}\, block > |file|$$

Communication: Communication involves maintenance and repair schemes.

$$|repair_{down}| = d \bullet |repair_{up}|$$

Computational complexity: The complexity can be computed for three main operations that are as follows:

For Insertion:

$$CPU_{(encoding)} = 5(k+h) \times n_{file} \times n_{block} \times I_{frag}$$

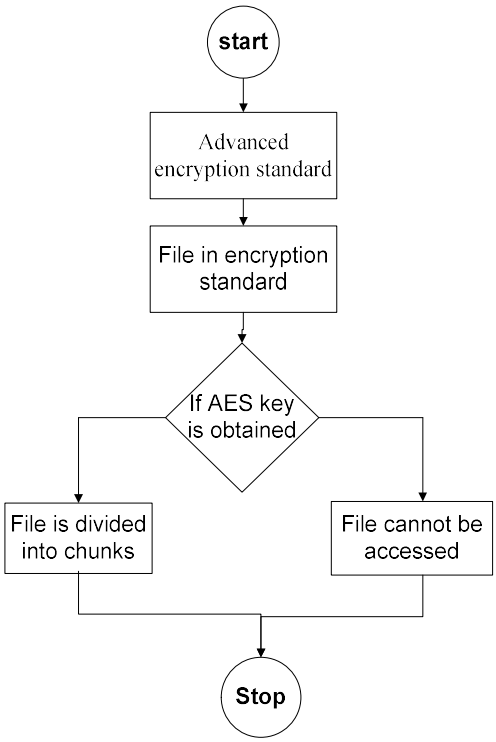

Fig. 4. Show encryption and decryption using AES key

For maintenance: $CPU_{(repair)up} = 5 \times n_{block} \times I_{frag}$

For reconstruction:

$$CPU_{(reconstruction)} = CPU_{(inversion)} + CPU_{(decoding)}$$

V. Formulation of AES

AES (Advanced Encryption Standard) uses an encryption key which is symmetric. It was implemented with the OPENSSL library. This includes the cipher block chaining mode, electronic code book, and cipher feedback. Security is one of the important features of the network structure. It can be achieved by one-way, weak collision-resistance, pseudo-randomness, or non-malleability by the usage of hash function.

In Fig. 4, encryption and decryption mechanisms are performed by using AES. The files uploaded by the user are divided into chunks, but only if AES key is obtained. Otherwise, the action is denied and user cannot access the file.

AES can be used in different types of modes, such as cipher block chaining (CBC), XTR, ECB, OFB, or CFB. In this model, we are implementing the AES algorithm in CBC mode.

In step 1, all the variables are initialized. In step 2 and step 3, input and outputs are described. In step 4, the AES key has been set up for the password. In step 5, the decrypted key is applied on the password. After using the decrypted key in step 6, the file can be accessed. In step 7, if the key is not applied on the password, then, in step 8, the file cannot be accessed.

A. Average Lifetime of Peers and Up Ratio

Average lifetime of peers can be calculated as $P_{LIFE.}$

$$P_{LIFE} = P_{ON} + \frac{1 - Pr_{death}}{Pr_{death}}(P_{ON} + P_{OFF}) \quad (2)$$

where

P_{ON} = Average session time (time spent by peer in online state).

P_{OFF} = Average disconnection time (time spent by peer in offline state).

P_{death} = Probability of peer to be dead.

Up Ratio can be computed as follows:

$$\beta = \frac{P_{ON}}{P_{ON}+P_{OFF}} \quad (3)$$

VI. Simulation Setup and Results

In this section, we validate that the Black-Box distributed peer-to-peer file storage backup is secured using the AES encryption algorithm. The validation process is conducted using MATLAB (matrix laboratory). In this program, we use symmetric shift keys for encryption and decryption. We use the GUI singleton MATLAB function for global point access and the GUI to call back for returning output arguments. We also use variable input and output arguments for the Caesar cipher encryption process. Table I below represents the parameters we used in our simulation.

In Fig. 5, the length of data is considered along the X-axis, and the Y-axis represents output data length for alphabet shift key 3.

In Fig. 6, the length of data is considered along X-axis, and the Y-axis represents output data length for alphabet shift key 4.

VII. Conclusion

In this paper, we present that, through the use of Black-Box, a distributed peer-to-peer network serves as an alternative for traditional storage devices for file storage and backup. The Black-Box system assures confidentiality and integrity by using an AES encryption key and various digital signature schemes. Using MATLAB, simulation results are produced and presented. Graphs are plotted with length of input data along X-axis and resultant output length along Y-axis, for respective AES key shifts. In addition, through the use of a symmetric AES key and cipher encryption cryptography, data confidentiality is greatly enhanced.

TABLE I: Parameters Used in simulation

Used parameters	Details of parameters
GUI	Global user interface
CBC	Cipher block chaining
X	Input data length
y_1	Output data length

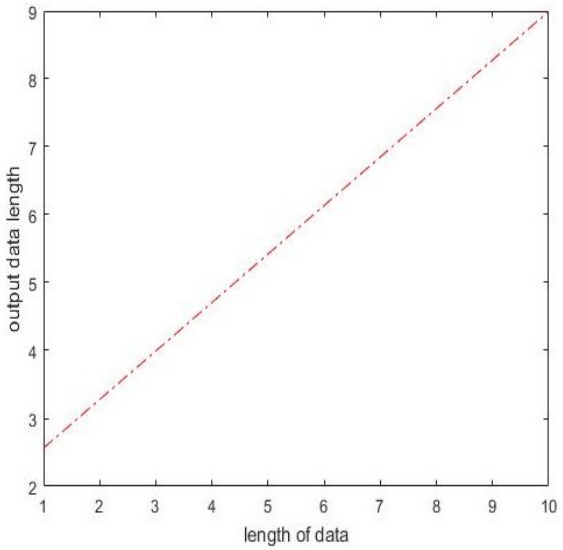

Fig.5. Input and resultant output length of data for AES key 3

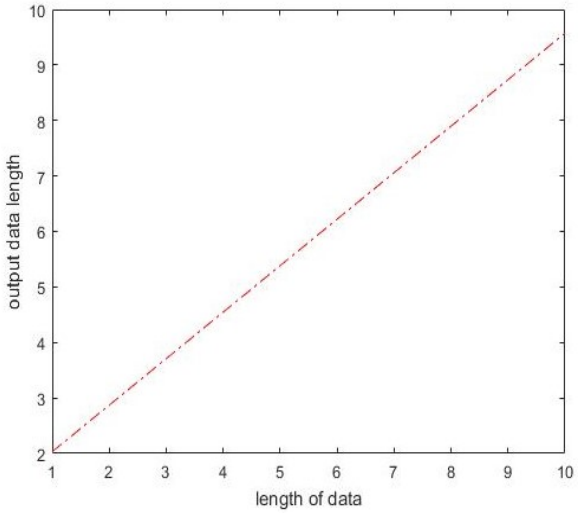

Fig. 6. Input and resultant output length of data for AES key 4

References

[1] G. Frédéric, J. Monteiro, and S. Pérennes, "Peer-to-Peer Storage Systems: A Practical Guideline to be Lazy," *In Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 2010)*, pp. 1-6, 2010.

[2] T. László, M. Amico, and P. Michiardi, "Online Data Backup: A Peer-Assisted Approach," *In Proceedings of 2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P)*, pp. 1-10, 2010.

[3] D. Abdulhalim, "Analysis and Optimization of Peer-to-Peer Storage and Backup Systems," PhD diss., Université Nice Sophia Antipolis, 2010.

[4] M. Julian, "Modeling and Analysis of Reliable Peer-to-Peer Storage Aystems," PhD diss., Université Nice Sophia Antipolis, 2010.

[5] Y. Zhi, J. Tian, and Y. Dai, "Towards a More Accurate Availability Evaluation in Peer-to-Peer Storage Systems," *International Journal of High Performance Computing and Networking,* vol. 6, no. 3-4, pp. 213-219, 2010.

[6] T. Yu-Chih, K. Ching-Ju Lin, and C. Chou, "Bandwidth-Aware Replica Placement for Peer-to-Peer Storage Systems." *In Proceedings of Global Telecommunications Conference (GLOBECOM 2011)*, pp. 1-5, 2011.

[7] R. Stefanie, B. Schiller, S. Hacker, and T. Strufe, "Measuring Freenet in the Wild: Censorship-Resilience Under Observation," In *Privacy Enhancing Technologies*, pp. 263-282. Springer International Publishing, 2014.

[8] C. Ian, O. Sandberg, M. Toseland, and V. Verendel, "Private Communication through a Network of Trusted Connections: The Dark Freenet." Network (2010).

[9] D. Peter, and A. Rowstron, "PAST: A Large-Scale, Persistent Peer-to-Peer Storage Utility," Proceedings of the Eighth Workshop on *In Hot Topics in Operating Systems*, pp. 75-80, 2011.

[10] A. Florian, S. Khayam, R. Jäger, and M. Rajarajan, "P2P File-Sharing in Hell: Exploiting BitTorrent Vulnerabilities to Launch Distributed Reflective DoS Attacks," In Proceedings of the *9th USENIX Conference on Offensive Technologies*, pp. 3-3, 2015.