International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 8 (2017) pp. 1703-1711
© Research India Publications. http://www.ripublication.com

Information Security System on the Basis of the Distributed Storage with Splitting of Data

Erbol Asylhanovich Kurmanbaev*, Iskendr Narimanovich Syrgabekov, Erkin Zadauly, Ardak Zhumagazievna Karipzhanova, Kumys Toleubekovna Urazbaeva

Kazakh Humanitarian Juridical Innovative University, Kazakhistan.
**Correspondence author*

Abstract

Background/Objectives: Current importance of the investigation has been stipulated by the urgent problem of safe data storage. The study aims to develop a new method of data protection different from encryption.

Methods: The principal approach to the investigation of the problem is represented by the method of data splitting within the distributed storage system that has been developed by the authors of the study. This system makes it possible to store the data preventing any unauthorized access. Separate split data do not carry any meaningful information. Another aspect is that the configuration of splitting/restoration can be designed in such a way that the data could be restored based on just a part of the split data; i.e. the data loss resistance can be ensured.

Findings: The commonly adopted paradigm of IT-resources holds a deep-rooted disadvantage represented by poor data protection from external intrusion. The study represents the results of the investigations of the principally new method of data storage employing the algorithms of splitting/reconstruction resistant against partial loss of storage areas. The method is founded on the concept of storing the distributed data not in special depositories but rather as a corporate cloud. The specific features of the system provide unique opportunities for security. First, the system administrator, the IT engineer and the relevant qualified technical personnel, who possess the full access to the system, cannot, nevertheless, access the data that are out of their terms of reference. Second, the responsibility for information security is handed over to the person competent in this specific area. In particular, the author/owner of the document commands the security parameters of his/her document at his/her own discretion without being concerned about physical safety of the information that is within the competence of the technical service. High integrity of the results of the investigation and, consequently, high degree of safety are confirmed by good adaptive, functional and dynamic properties revealed by the information safety system based on the distributed storage technology employing data splitting on its introduction to cloud.

Applications/Improvements: The study is of practical value for all IT systems including cloud technologies. The developed algorithm enables the implementation of the internally consistent model of safe data storage and processing.

Keywords: Distributed Storage, Safety, Data Splitting, Cloud Technologies

INTRODUCTION

Cloud technologies are gaining ever greater popularity as they are convenient and attractive for the users who deal with the large-scale calculations and have to store large arrays of data. Cloud technologies enable the companies to reduce costs and improve revenues employing the inexpensive services. These trends can drastically change corporate structures and social life, as the opportunities of cloud technologies are seemingly limitless. There is a huge year-on-year growth of users. According to Gartner, by the end of 2015 the spending on cloud technologies across the world made over USD 180 billion [1]. The investigation project represented by Goldman Sachs in the beginning of 2015 says that the year average growth rate of spending on the infrastructure of cloud calculations and platforms would make 30 % by 2018 [2].

However, the initially revealed vast prospects have already been compromised by the facts of unauthorized entries. The most trivial example is the massive hacking of iCloud server user accounts detected on August, 31, 2014 when hundreds of private pictures of celebrities appeared on the Internet. It is believed that the reason for the leak was the vulnerability of iCloud server that synchronizes iPhone pictures, although Apple denies iCloud hacking.

And this was not only one occasion. Therefore, the users all over the world are wary of the growing threats. No one wants any unauthorized access to one’s personal or corporate data, not to mention their loss. It is generally believed that the information in public clouds cannot be properly protected. Consequently, the latest trend is to move to private (corporate) clouds. However, this is practically useless, because today almost every system has access to the Internet and, even under the conditions when the server is “locked” during the relocation of the applications and databases into the private cloud, the safety of the data cannot be guaranteed. Besides, in this case there is no access to much larger data processing facilities and to super storages that could be provided by the public cloud.

The problem of security is the principal barrier faced in the course of implementation of cloud technologies not only in companies and corporations. In governmental sector this issue is of much more greater importance. According to Brookings Institution, the lack of security is the major obstacle that prevents the US Federal Government from shifting more functions into the cloud platforms in future. Moreover, the government of Kazakhstan had to pass Resolution No. 965 dd. September, 14, 2004, On Some Measures for Ensuring the Information Security in the Republic of Kazakhstan, according to which "processing and storage of the data that represent state secrets and restricted access information should be performed with the computer equipment that is not connected to international (global) data transfer networks, Internet and/or information networks, communication networks that have access to international (global) data transfer networks, Internet" [3].

Is it possible, under these conditions, to improve the reputation of public clouds? The only way to do this is to ensure the reliable safety of the stored data. Poorly manageable infrastructures of cloud data storage pose threats before any business, any governmental enterprise in case of catastrophic failures. Consequently, the principal task is to create the economically efficient architecture featuring extremely safe storage capacity, scalability and seamless integration between the local and cloud-based methods of data storage.

The objective of this study is to solve the issue of cloud system security. The essence of this project is to employ the innovative distributed split data storage technology developed by the authors of this study and tested within the local networks. It seems to be high time for introducing innovations into the cloud systems.

Within the framework of the suggested technology the new class algorithms split the data into a large number of files; thereat, each of these files cannot contain even one bit of the initial information. The split files are distributed among multiple servers programmed for self-reconstruction and self-preservation which ensures continuous safekeeping and security of the data.

The system cannot be deciphered, because it does not represent encryption. It can withstand any possible attacks. Multilayer security structure and the unique resistance against the destruction of the storage areas ensure guaranteed protection from most sophisticated hacking. Due to the innovative hierarchical access protocol, the system prevents any unauthorized access to the data and any potential thefts by the insiders. Finally, the system ensures 100 % safe data transfer through the open Internet channels and private networks.

Solving the issue of public cloud protection the authors of this study also help solve the relevant problem in the global context given the unity of the virtual space.

CONCEPT HEADING

The existing structure of IT resources arrangement has been historically founded on the current paradigm of local calculations focused on local resources [4]. Thereat, traditional approach implies that the resources should be centralized into powerful specialized clusters to boost the quality of IT solutions [5]. From the perspective of organizations it represents specialized server platforms in the company, standalone data centers [6] or specialty services provided by third party organizations to meet the existing demand for IT services. However, localization and centralization of the resources also possess "inherent" disadvantages that start affecting the whole further development of IT industry.

The major disadvantage of the commonly adopted platform is its low level of data storage reliability. Thereat, this is true for all aspects of informational security: storage reliability, leak protection and resistance against the distortion of data [7]. Obviously, if no timely measures are taken, the poorer information reliability can lead to catastrophic consequences. And, of course, this is due to the general trend to move all information resources to computer storage media, due to the deep penetration of computer technologies throughout all spheres of human activities. Moreover, today large amounts of information are stored in the external cloud systems, i.e. in large network depositaries accessed through the Internet. Private information in cloud systems is protected from interference; however, the external cloud systems have already been hacked more than once. Indeed, the companies that set higher requirements to information security (governmental, banking, commercial structures, state-owned companies, etc.) prefer to store their data in their own expensive (maintenance costs including) depositaries that in fact represent specialized data centers.

Given the abovementioned trends, the increased demand for storage facilities is accompanied by generally decreasing reliability of hardware platforms. Nowadays, the following facts can be observed: a) a rapid increase in the record density associated with a simultaneous decrease in reliability: a single failure of the hard drive can result in the loss of catastrophic amounts of information; b) approximation of the parameters of the expensive professional level hard drives to the parameters of inexpensive drives for domestic applications, and, consequently: c) application of cheap drives and drive subsystems (RAID arrays that are still sustainable even in cases when one of the drives fails) in the servers of middle and initial levels.

Of course, cloud system program developers employ ever more original protection methods. To begin with, the hacking protection systems are constantly improved. Nevertheless, this battle can last forever; thus, it would be advisable to return to the very beginning and to look for the new methods of storing large arrays. And such methods do exist. Thereat, cloud technologies do not have to be necessarily abandoned. It would be enough to change the ways of data storage.

The system of distributed information storage developed by the authors of this study is meant to improve the database security. It employs the principle of distributed storage applying the algorithms of splitting/reconstruction of the storage areas that are resistant against partial losses. The described system is founded on the concept of distributed data storage within the local network that represents the corporate cloud. That is to say that this investigation aims to create the programs to arrange data storage not in specialized depositaries but in the computers of the internal network of the company. Actually, here there is also a cloud, but it is the corporate cloud featuring the depositary extended to the network computers and employing proprietary software. This security system can also be easily applied to public clouds.

This is a completely new approach to information safety unrivaled in the world of information technologies. Its principal differences from the existing options are represented by the very idea of distributed storage in the corporate cloud of the internal computer network that can function even in cases when some computers that function as storage areas are disconnected and also by the possibility to avoid cumbersome ciphering algorithms, because the suggested system is not encryption.

METHODS

Modern information flows primarily tend to store large arrays of data in the distributed cluster systems. Conventionally, they can be subdivided into two classes as follows: distributed file systems (Google File System, Hadoop Distributed System, etc.) and distributed structured data storages (Google BigTable, HBase, etc.). These systems are principally different from the traditional file systems and relational data bases. For example, distributed file system Google File System is a proprietary design of Google applied for storing big arrays of data. Inside Google, there are over 200 GFS-clusters of which the largest include more than 5 thousand units that store circa 5 petabyte of data and serve circa 10 thousand clients [8]. As well as any other distributed file system, GFS aims to ensure high efficiency, scalability, reliability and availability. However, the security problems are still there.

By contrast to such complex systems, the innovative system of distributed storage developed by the authors of this study is founded on splitting of data. This method of storage has several aspects that evoke interest. As for the first of these aspects, the data splitting distributed storage systems make it possible to store the data and prevent the unauthorized access to the information. Separate split data do not possess any meaningful information. The second aspect is associated with the fact that the configuration of splitting/reconstruction can be designed in such a manner that the data could be restored based only on a part of the split data and thus the data loss resistance could be ensured.

The principal scientific method in the systems of improved reliability and safety is represented by the application of complex mathematical algorithms that realize the procedure of secret sharing [9, 10]. In cryptography secret sharing is understood as any method of secret distribution among the group of participants where each participant is provided with his/her share of the secret. Thereat, only the coalition of the participants can recreate the secret. In order to ensure sustainability in cases of information loss, the threshold scheme is applied when the number of the shares needed to recreate the secret is lower than the number of the shares into which the secret was divided [11].

The most widely applied are the algorithms based on Adi Shamir's schemes [12], and, in particular, the well-known Reed-Solomon code (which, for instance, is applied in platform Wuala) [13]. The major disadvantage of such algorithms is that they require more and better computational resources, because the implementation of the algorithms needs complex mathematical calculations and lacks scalability [14].

There are alternative secret sharing schemes that are based on even more complex mathematical models. For example, Blakley's scheme [15] operates with *n*-dimension hyperplanes; and Karnin-Greene-Hellman scheme [16] makes use of the mathematical theorem that says that it is impossible to solve the system of *n* equations when there are *m* unknown values. Given these disadvantages, the systems founded on secret sharing schemes gained little practical application, and it seems probable that the abovementioned problems are the principal factors that predetermine slow development of these systems [17].

Special attention should be paid to the method of accessing the data of the computing unit within the distributed computing system featuring the dispersed storage network that ensures the improved efficiency at the level of the system by means of storing metadata and data parts within one single complex in module DSTN (dispersed storage and task network) [18].

At the same time, simple and fast parity algorithms have been widely applied in the correction algorithms. In particular, they are applied in the widely spread systems of redundant arrays of independent disks (RAID) for servers. On the other hand, the fast and simple parity algorithms cannot retrieve the multiple data losses. For instance, in RAID systems only one disk in the array can fail.

The distributed storage system developed by the authors of this study employs the proprietary parity algorithms resistant against multiple failures. The implementation of the system covers a number of patents from Great Britain and the European Union along with the know-how of the authors of this study. The target platform for the system is MS Windows XP/7/8/8.1 with .NET Framework v.4/4.5.

RESULTS

System Operation Principle

The model of distributed or grid computing proper is not, in fact, an innovation in IT sector. In the first computers this model was regarded as a promising trend. However, it failed to withstand the competitive struggles against the fast developing technologies of semiconductor engineering under the conditions of the sharp increase in local computer capacities. Nevertheless, at the new spiral turn of computer technology development associated with faster data transfer and with the ubiquitous implementation of broadband wireless and optical channels, with emerging new algorithms and ideas in the sphere of distributed computing the model of resource distribution now makes it possible to enter the qualitatively new level of information processing.

The developed and investigated system of distributed data storage makes it possible to store and to process the client data at the distributed nodes. The data are divided in parts by the algorithm and are distributed to the nodes of the system.

The system consists of two functional components: “Client” and the distributed array represented by the network of mutually interacting “Nodes”. “Client” (hereinafter Node and Client without inverted commas) is a computation unit represented by personal computer, server or any other smart device with installed “Client” software. Client serves as a gate for entering the system. Node also can be represented by any computer unit with installed “Node” software.

Client operates with the system as with an abstract cloud subsystem. The interaction between the Client and the array of Nodes is performed through one-to-many protocol. Any act of information recording in the system undergoes preliminary treatment with special algorithm which: a) splits the information into unreadable component parts; b) adds dynamically generated redundant data to improve sustainability in cases when the split parts are partially lost; c) generates utility metafile that describes the created array. The act of recording the set of the formed data is performed by way of redistributing them to the nodes of the system. Any act of reading the information by the Client in the system is only possible by way of processing the array of the data that have been obtained at the nodes through the algorithm capable of reconstructing these data based on metadata only.

The nodes of the system that form the distributed array know only limited number of their neighbors. None of the nodes can know the whole system and the nodes it is made of. The nodes can connect to and disconnect from the system dynamically without affecting the operability of the system as a whole. The nodes can exchange data and automatically update the lost parts of the stored information on the commands from the Client.

The Client can pass authorization at any node of the system in order to operate with the whole system. All information interactions between the components of the system are performed through the channels that represent virtual tunnels. The system of distributed data storage makes it possible to store and to process the client data at the distributed nodes. The data are divided in parts by means of the algorithm and are distributed among the nodes of the system.

The system can withstand collective disconnections and damages of the nodes up to 50 % of their full capacity depending on the dimensions of the system and on the configuration parameters of the algorithm. Separate nodes store the information that has no functional meaning.

The system does not require certification in terms of the use of crypto-algorithms, inasmuch as it does not apply encryption to protect the stored data. The data can only be read upon being reconstructed from the data that are “spread over” the system at the Clients. The data can be reconstructed only by the owner (creator) or by anyone authorized to do so. Authorization does not mean the transfer of ownership. The owner always keeps under control any changes in his/her files.

Thus, the specific features of the system are as follows:

1) Anonymity. Client data are split in blocks and are recorded at different nodes of the system which ensures anonymity and makes it possible, without applying any encryption methods, to protect the client data from unauthorized access. The data split in such a manner and stored at different nodes of the system can only be processed by the owner.

2) Resistance against data loss. The splitting algorithm enables data restoration even when 50-80 % of the nodes of the system are lost.

3) Minimal requirements to the nodes of the system. All principal data-related operations are performed by the Client; thus, the requirements to the nodes that store the data can be reduced.

4) Enhanced functionality. The system does not just store data; it enables the connection of the modules as follows: mail service; Voice over IP; single access to the same resource for several users.

System Architecture

The architecture of the application is server-client. The system has been developed on Visual Studio 2010 with programming language C# employing .Net technique.

The system consists of the nodes interconnected through the trusted communication channel and of Clients that cooperate with those Nodes (Figure 1). Each node knows only the neighboring nodes and does not understand the system as a whole. The node keeps the data of the Clients in Datafile. The incoming information is processed in the order of arrival.

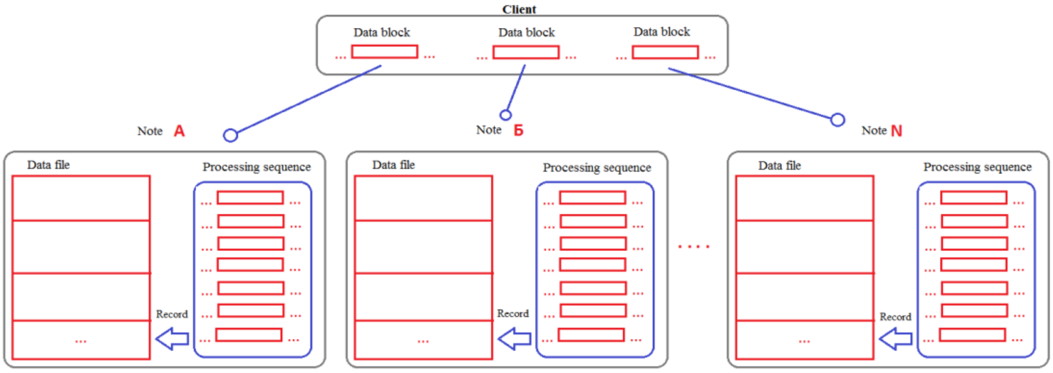

Figure 1. Schematic architecture of distributed information storage system

The Client gets access to the nodes of the system only upon registration within this system. The client is provided with a unique key that is used in the process of data distribution. The Client does not have information about the rest of the nodes in the system.

The Client application is in charge for splitting the file into the blocks and for distributing the blocks among the nodes. For the purposes of storing the file system and distributed blocks, the client application uses metafile (or mf) (Figure 2). Due to its tree-like system, the metafile describes the file structure and the arrangement of the data blocks.

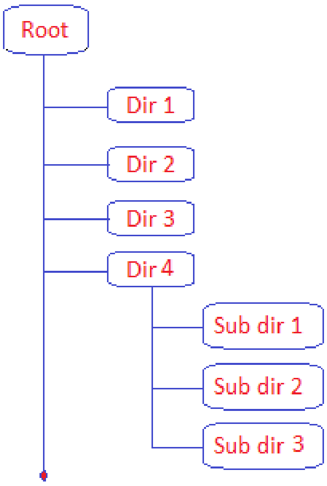

Figure 2. Metafile structure schematics

When the operations are accomplished, the metafile itself is divided into blocks and distributed between the nodes. Only the unique key of the Client (CID) in combination with the authorization data (user account, password) can restore the metafile.

The user installs the client part of the application and gets authorized in the system obtaining the unique key or CID which serves to create the metafile. Upon selecting the required file, the Client starts the function "Upload to the cloud". Thereat, this file is blocked and, in the background, is divided into blocks and distributed among the nodes of the system.

If the client wants to access his/her own file, he starts the function "Download from the cloud" and receives the data blocks of his file from the cloud; these blocks are compiled into a single file following the algorithm. The client can grant access to his/her directory to other users.

Application Description

Server (back end) part of the application serves to store and process the information that comes from the client. The data are stored in the data files that are split into clusters. The cluster, in turn, is split into the pages ranked according to double multiplex system. The size of the cluster can be configured using the settings of the system; the cluster itself contains the number of pages multiple of two. For instance, if the cluster consists of 32 megabytes, then it contains 256 pages sized 128 kilobytes each. As new information arrives, the system creates new clusters to store the data with different amounts of pages.

All incoming data are enrolled into the queue which controls all the processes of data recording. Upon successful reception of the client data block, the system records this block into the data file and sends a message to the client notifying the successful reception of data block as UID.

The nodes are interconnected through the trusted communication channel used for data exchange. For example, when the new user is introduced into the system, the node informs the neighboring nodes about the creation of a new user.

Server part of the application performs the functions as follows: a) authorization of the client (incoming parameters: authorization data; resulting parameters: SSKey, Root_ID); b) client file storage (incoming parameters: data block without UID; resulting parameters: UID); c) client data block transfer (incoming parameters: UID; resulting parameters: data block); d) echo request (incoming parameters: echo request; resulting parameters: IP address); e) neighbor questioning (incoming parameters: inquiry to neighbors; resulting parameters: Node ID); f) data exchange between the nodes.

When the new user is registered, when the existing user is excluded or when the user data are modified, the nodes will exchange this information.

The client (front end) part is the software installed on the computer of the client that enables client's registration within the system and that manages the data blocks and files. The client part should perform the functions as follows: a) file splitting (incoming parameters: data file; resulting parameters: data blocks); b) file reconstruction (incoming parameters: data block; resulting parameters: file or error message); c) data sending (incoming parameters: data block; resulting parameters: UID); d) data reception (incoming parameters: UID; resulting parameters: data block); e) authorization (incoming parameters: UID; resulting parameters: data block).

The elements of the interaction between client and server parts: a) authorization (incoming parameters: UID; resulting parameters: data block); b) data reception (incoming parameters: UID; resulting parameters: data block); c) data sending (incoming parameters: data block; resulting parameters: UID).

System Programming

Principle results of programming the distributed storage system with splitting of data are as follows: 1) required nodule specifications; 2) results of the component interaction chart design and testing; 3) component interaction protocols; 4) system module programs; 5) system component modules and module interface programs; 6) server component programs; 7) storage node programs; 8) user interface design programs; 9) program system at pre-alpha stage; 10) functionality testing and finalization results; 11) results of hardware component interaction tests and recommendations on application; 12) alpha-version final configuration; 13) system with services installed in the local cloud; 14) preliminary system testing results.

The distributed data storage system on the basis of splitting of data technology employs module architecture. Therefore, in general, the program represents a set of the components (modules) and the set of built-in program interfaces for those components which enable the creation of flexible program architecture while ensuring the simplicity of design. The authors of this study believe that it was the creation of the module system that predetermined the required properties of the corporate cloud. Indeed, notwithstanding any future changes in the requirements, the flexibility attained due to separate and independent components will make it possible to change/add/remove the functions "on-the-run" causing no delays in current operations, and also, consequently, to create new opportunities. In principle, the operations of any module can be represented as schematics shown in Figure 3.

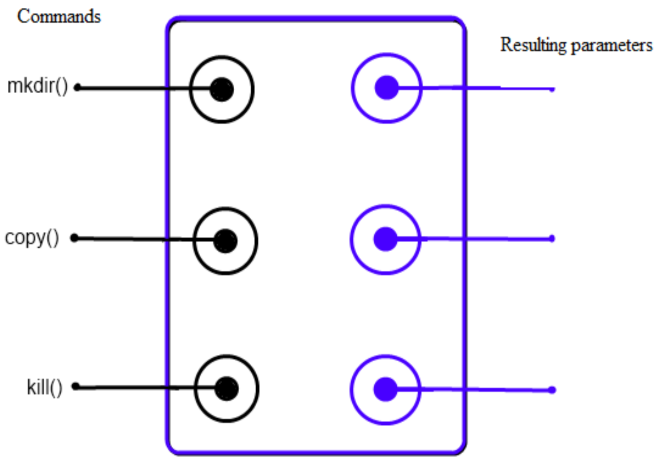

Figure 3. Module operation schematics

In all developed modules the incoming parameters (on the left in the schematics) are represented by the commands: a) from other modules; b) from the users, sent, for instance, by clicking the key on the screen; c) network commands, etc. All commands have been described by the programmers within the interface of the module. The resulting parameters (on the right in the schematics) are represented by the data flows or, in a number of other cases, by the commands to launch other modules. The modules interact with the external environment and with each other through program interfaces. The modules can be built in one another.

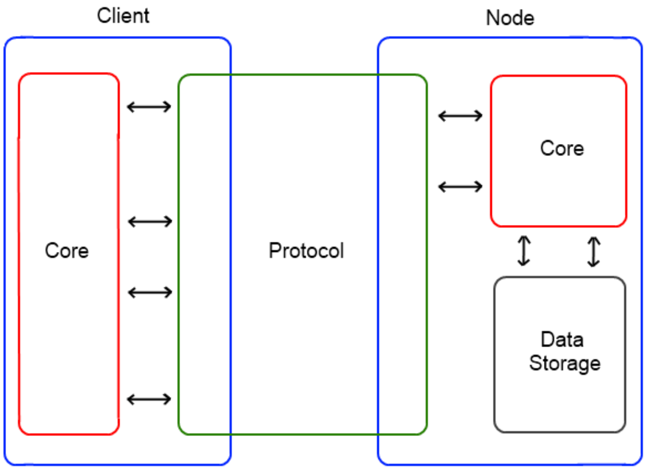

Figure 4. Module interaction schematics

The joint operation of all modules has been shown schematically in Figure 4. Here, Core is the module that realizes the algorithm of data splitting in the client application and the algorithm of data reconstruction in the node application after splitting. Node is the module that realizes the algorithm of storing and distributing the data at the nodes of the system. Client is a module that is in charge for the interaction with the distributed storage system; it provides the possibility for the final user to work with convenient and friendly interface of Windows. Several modes of operation have been envisaged: constantly operating background service (daemon in Linux terminology), interactive program. Protocol is a module that performs the functionality of Client-Node interaction. Data Storage is the module that stores the data at the nodes of the cloud. The schematics show that the architecture of the distributed data storage system on the basis of splitting of data is simple and reliable.

The interaction between Node and Client is performed through the program module Protocol (Figure 5).

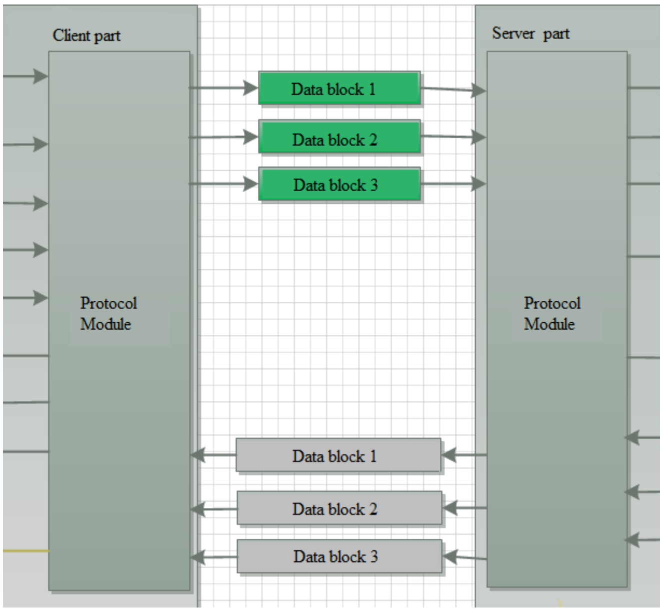

Figure 5. Schematics of Protocol module operation

The program interface of Protocol module provides the possibility for other modules to get connected to it, and, using the set of standardized typical commands (API), to exchange information with each other and with “outer world”. This module is crucial for the distributed data storage system; it is present in all components of the system.

One of the most important components of Node module is the Fst internal data storage module. It consists of the components as follows: a) data block recording module; b) data block reading module; c) data flow control module; d) data storage module.

Module Node Net is a network module which establishes safe transactional model of data transfer within the system of the most reliable data storage.

Module Manager is in charge for compiling the data blocks, for generating the unique keys and for redistribution of the blocks at the storage areas. The module makes use of the unique properties of the algorithm to improve storage reliability, especially its capability to generate sustainable clusters and clusters of clusters. Module Node Net employs the principles of distributed network Kademlia to evenly distribute the data blocks within the net. The algorithm has been designed as Core module.

Admin is the module of the administrator; it serves as the key link in the system to manage the users. Also, module Admin makes it possible to redistribute the data flows in the net manually, in order to reduce load on the most active storage areas.

Module Storage Service. This module has one important task, namely, “physical” storage of data blocks at the nodes of the network.

The arrangement of the modules is designed in such a way that, in order to ensure sustainable and safe operations of the distributed data storage system on the basis of splitting of data, all modules in the system are connectred with each other through program interface API (Application Program Interface); i.e. each module has the set of incoming and resulting parameters of its own. The resulting parameters are, as a rule, represented by sets of data, by the results of module operations.

The interaction between the modules is ensured by the program package. Figure 6 schematically shows the interactions between all modules available in the distributed data storage system based on data splitting technology.

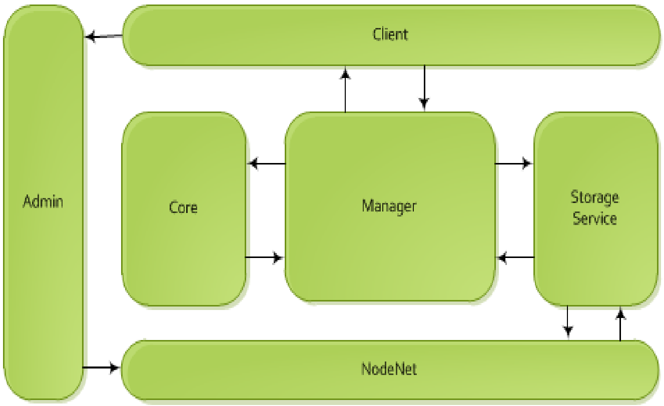

Figure 6. Module interaction schematics

The picture illustrates that some modules are isolated because of their narrow specialization.

To enable user operations, the system requires graphical interface that, in its essence, is just a “skin” for the majority of the modules in the system. The basic task of the graphical interface is to make user operations as easy as possible and to improve the protection from users’ indiscrete actions. The graphical interface or GUI (Graphical user interface) is present within Client module and represents the comfortable environment similar to Explorer. The combination of GUI and systemic service enables fast visualization of the most important user information saving the one from having to launch multiple modules manually.

Comparative Aspect

There are no more than several examples of how the idea of distributed data storage can be realized: platform Clever Safe (distributed storage at the servers of the company using hard- and software complex on the basis of iSCSI and proprietary algorithm) [19], recently emerged Symform (technology Symform Cooperative Storage Cloud that uses client computers as parts of the system and that also employs RAID-96 which is a proprietary option of RAID) [20] and Wuala (technology similar to Symform that uses Reed-Solomon code) [13]. All these systems offer the services of distributed storage at their servers and employ different splitting/reconstructing algorithms suggesting uploading the data through the Internet to the servers of the company. To protect the data from unauthorized access, encryption methods are usually applied. These services are provided on commercial basis and require the installation of specialized hardware (Clever Safe) or they can require the provision of some extra services, for example, allocating the free space on the computer drive which is supposed to be used by the system for its own purposes (the drive acts as a component part of the distributed storage system) and/or providing constant access to the computer through the Internet from the outside.

A standardized system for ensuring storage reliability and safety can be exemplified by portal aws.amazon.com: cloud storage service S3 (Simple Storage Service) and EBS (Elastic

Block Store) [21] that apply technologies of clusterization and replication.

The investigated system developed by the authors of this study, by contrast to the existing distributed data storage systems, suggests for the first time ever that the distributed storage system should be designed as a part of corporate cloud. The available counterpart distributed storage systems (Clever Safe, Wuala, Symform) envisage data uploading to the server of the company through the Internet [22,23]. Thereat, the distributed storage system developed within the framework of this study suggests that the data should be stored directly in the computers of the internal network.

System Testing

The distributed data storage system based on data splitting technology has been tested in the local cloud with 16 units of computer equipment. Initially, the system has been load tested and showed stable results operating with data up to 2 GB. Then the distributed data storage cloud system was tested in the internal network of the university. Within the framework of the tests the correlations between the data processing speed (which is the float value that totally depends on the parameters of the computer equipment) and the scope of the processed data have been measured. The tests were carried out for data amounts of 1 MB, 100 MB, 300 MB and 600 MB.

The analysis of the results showed that the tests proved stable operation of the system under real loads up to 2 GB. The parameters of stabilization can be estimated using the data processing graphs. For instance, processing the data of 1.5 GB takes circa 14 minutes. The threshold values of data downloading and uploading are equalized and become approximately equal. Similar parameters were observed when the speed of data processing in the cloud was changed.

The test investigations enable positive conclusions that the distributed storage system based on the data splitting technology operates sustainably within the cloud structure. The abovementioned system parameters that result from the specific features of the data processing algorithm realize the new paradigm in the sphere of computer security. Generally, it can be described as a model where close but not identical characteristics of the stored information are divided. This is just to say that data are usually understood as two different entities: a) physical data: bits, files, hard drives with files; b) information proper (semantic contents), the message of the Word document text, a man’s face in the digital picture, characteristics of the device reflected in the drawing, etc. Thus, the notion of information security indeed covers two absolutely different contexts. In the described distributed storage system these two contexts coexist separately and lend themselves to independent regulation.

CONCLUSIONS

In all, the specific features of the system predetermine the unique properties that help ensure security and that imply the realization of two contexts. First, the system administrator, the IT engineer and the relevant qualified technical personnel are in charge for data safety. Nevertheless, possessing the full access to the system they cannot access the data that are out of their terms of reference (the data in the sense of informative contents). Second, the responsibility for information safety rests with the person competent in this specific area. Naturally, the author/owner of the document should command the security parameters of the document at his/her own discretion without being concerned about physical safety of the information that is within the competence of the technical service. Thus, it can be concluded that the integrity of the results and, consequently, the high degree of safety are proven by good adaptive, functional and dynamic properties demonstrated by the system based on the distributed storage technology employing data splitting on its introduction to cloud.

REFERENCES

[1] About some measures for ensuring the information security in the Republic of Kazakhstan. Resolution No. 965 of the Government of the Republic of Kazakhstan dated September, 14, 2004. http://adilet.zan.kz/rus/docs /P040000965. Date accessed: 08.09.2016.

[2] Adi Shamir. How to share a secret. Communications of the ACM. New York, ACM. 1979, 11, 22, 612-613. ISSN: 0001-0782.

[3] Asmuth C, Bloom J. A modular approach to key safeguarding. Information Theory, IEEE Transactions on. 1983, 2, 29. ISSN: 0018-9448.

[4] Blakley GR. Safeguarding cryptographic keys. Proceedings of the 1979 AFIPS National Computer Conference. NJ: AFIPS Press, 1979, 313-317. DOI: 10.1109/AFIPS.1979.98.

[5] Boglovskaya E. The company “Servionica” announces the beginning of cooperation with new resident of technopark “Innopolis”. SaaS.Ru. 2015. 1 July. http://saas.ru/posts/~siervionika-rasshiriaiet-komandu-razrabotki-riesursami-novogho-riezidienta-innopolisa. Date accessed: 07.10.2016.

[6] Comparison of encryption schemes. http://www.wuala.com/en/learn/technology. Date accessed: 08.05.2016.

[7] Dean J. Handling Large Datasets at Google: Current Systems and Future Directions. Data-Intensive Computing Symposium, 2008. http://research.yahoo.com/files/6DeanGoogle.pdf. Date accessed: 07.12.2016.

[8] Haratishvili D. Data centers in facts and figures. http://www.sapr.ru/article.aspx?id=20687&iid=942. Date accessed: 07.11.2016.

[9] How Cleversafe Works. http://www.cleversafe.com/overview/how-cleversafe-works. Date accessed: 08.05.2016.

[10] Karnin E, Greene J, Hellman M. On secret sharing systems. Information Theory, IEEE Transactions on. 1983, 1, 29, 35-41. ISSN: 0018-9448.

[11] Mignotte M. How to Share a Secret. Lecture Notes in Computer Science. 1983, 149, 371-375. ISBN-10: 3-642-11446-6.

[12] Mishechkin A. The cluster differs from the other cluster. Windows IT Pro. 2008, 5. http://www.osp.ru/win2000/2008/05/5529265. Date accessed: 08.11.2016. [13] Monika G, Kalpana Y. Data Security is the Major Issue in Cloud Computing – A Review. IJST, 9, 43, November, 2016.

[14] Novikov YuV, Kondratenko SV. Fundamentals of local networks. Moscow, Internet-University information technology, 2005. ISBN 5-9556-0032-9.

[15] Partyka TL, Popov II. Information security. Moscow, Forum-Infra-M, 2004. ISBN 5-8199-0060-X.

[16] Popsulin S. The global cloud services market will surpass the IT market dozens of times. CNews.Ru.http://www.cnews.ru/news/top/2016-01-25_mirovoj_rynok_publichnyh_oblachnyh_uslug_v_desyatki. Date accessed 01.25.2016.

[17] Resch JK, Leggette W. Method for accessing e.g. data of computing device in dispersed storage network involves generating access specific key based on content specific information and executing access request regarding data object utilizing access specific key. Patent Number: US2014068259-A1. [18] Satish Kumar, Anita Ganpati. An Approach for Data Security from Malicious Attacker in Cloud Computing. IJST, 9, 32, August, 2016.

[19] Schneier B. The algorithms of secret sharing. In: Applied cryptography. Protocols, algorithms, and source code in C. M., Triumph, 2002. 588-591. ISBN: 5-89392-055-4, 0-471-11709-9.

[20] Schneier B. The secret sharing. In: Applied cryptography. Protocols, algorithms, and source code in C. M.: Triumph, 2002, 93-96. ISBN: 5-89392-055-4, 0-471-11709-9.

[21] Security Resources. http://aws.amazon.com/security /security-resources. Date accessed: 08.06.2016.

[22] Simmons CJ. An introduction to shared secret and/or shared control schemes and their application. Contemporary Cryptology. IEEE Press, 1991, 441-497. ISBN: 0-87942-277-7.

[23] The Smartest Cloud Security. http://www.symform.com/how-it-works/security. Date accessed: 08.05.2016.