# About Cloud Storage Systems Survivability

Nikolay Kucherov [a], Inna Dvoryaninova [a], Mikhail Babenko [a,b], Natalia Sotnikova [a] and Nguyen Viet Hung [c]

[a] *North-Caucasus Federal University, 1, Pushkin Street, Stavropol, 355017 Russia*
[b] *Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*
[c] *LeQuyDon Technical University, 236 Hoang Quoc Viet, Hanoi, Vietnam*

**Abstract**

This article proposes an approach to improving reliability and survivability based on modular arithmetic. The proposed approach makes it possible to increase the survivability of cloud storage systems, as well as reliability and fault tolerance of data storage. To increase fault tolerance in the event of a failure, the redistribution of the processed data is applied. The proposed model allows restoring the saved data in the event of failure of one or more cloud servers.

**Keywords** [1]

Cloud computing, error correction code, survivability

## 1. Introduction

As the amount of stored data increases, more and more users are switching to cloud storage systems. In modern conditions, the requirements for the reliability of data storage in cloud are constantly increasing. The reliability of cloud storage [1-4] is understood as its property to ensure the management and storage of data while maintaining the values of the established quality indicators over time in operation. It reflects the impact on the performance of cloud storage mainly of intra-system factors - random failures of technology.

Cloud storage survivability [5-8] means its stability of the control and transmission system against external causes, aimed at disabling cloud storage, as well as resistance to cascading failures.

The concepts of reliability and survivability have much in common and at the same time differ significantly from each other. They are united by the principle of stability, which takes into account all the variety of factors, including various emerging failures. The stability index is a function of the reliability indicators, survivability and fault tolerance.

Differences in the concepts of reliability and survivability and reasons of normal cloud system functioning disruption are due to significant differences in their manifestation, the nature and scale of failures, its duration, methods of their elimination and methods of increasing fault tolerance. Accordingly, the initial data, calculation methods, accuracy and the essence of reliability and survivability indicators differ significantly. The first ones are well provided with statistical material, the main influencing factors are taken into account, they are deeply developed theoretically, and they can be used for sufficiently accurate forecasting, calculations, design and modeling. The group of survivability indicators more reflects the qualitative picture of the behavior of the considered cloud storage system in conditions of external influences or cascade propagation of failures [9, 11].

The reliability of cloud storage systems manifests itself in the form of failures. The concept of failure is closely related to the concept of operability. Operability is the state of the cloud system, in

YRID-2020: International Workshop on Data Mining and Knowledge Engineering, October 15-16, 2020, Stavropol, Russia
EMAIL: nkucherov@ncfu.ru (Nikolay Kucherov); innadv99@mail.ru (Inna Dvoryaninova); mgbabenko@ncfu.ru (Mikhail Babenko); sotnikova-natali@list.ru (Natalia Sotnikova); hungnv@mta.edu.vn (Nguyen Viet Hung)
ORCID: 0000-0003-0337-0093 (Nikolay Kucherov); 0000-0003-2174-2284 (Inna Dvoryaninova); 0000-0001-7066-0061 (Mikhail Babenko); 0000-0001-5029-0390 (Natalia Sotnikova); 0000-0002-9818-4455 (Nguyen Viet Hung)

which it is able to perform the specified functions with the parameters and requirements for the quality of the services provided. Failure - a random event that disrupts the performance of the cloud system.

A variety of approaches can be used to build a distributed system for storing and processing data. Some of them are based on cloud computing paradigms. These infrastructures have both common characteristics and fundamental differences. Using clouds for storage requires security, reliability and scalability with limited Internet bandwidth to provide fast access to distributed data and a high degree of reliability, availability and scalability.

Distributed storage can be based on multiple clouds. Typically, data is divided into several parts, which are stored in different clouds to ensure availability in the event of a failure. However, failures in distributed storage can cause inconsistencies between different copies of the same data.

Large databases can be used. In this case, to ensure high performance, data processing and analysis must be performed using parallel computing.

In the second part, the main problems of cloud storage and the uncertainty of the emerging failures are considered, in the third part, methods for increasing the reliability and survivability of cloud storage are given and the application of the residue number system for this problem is considered. The final part is about the method of increasing reliability as information backup, its advantages and disadvantages are also given.

## 2. Problems of cloud storage systems

When storing data in the cloud, situations arise in the denial of access to data, errors or deterioration in the functioning of services, and sometimes in long interruptions in their work. For these and other reasons, distributed data processing is inevitably a continuous stream of failures, errors and malfunctions. Cloud storage failure can occur slowly, over a long period of time, or in a split second.

Uncertainty can be seen as the difference between available knowledge and complete knowledge. It can be classified in several different ways depending on their nature. [4].

In the implementation of the cloud storage parts, the occurrence of failure uncertainty can be qualified as follows:

Uncertainty of software failure is a limited uncertainty due to complete or partial ignorance of the conditions under which decisions must be made.

The uncertainty of a hardware failure, such as hard drives, power systems, etc., is a technical uncertainty and is a consequence of the inability to predict the exact results of solutions.

Information theory founder Claude Shannon defined information as removed uncertainty. More precisely, obtaining information is a necessary condition for removing uncertainty. Uncertainty arises in a situation of choice. The problem of reducing number of options under consideration (reducing the variety) and, as a result, the choice of one corresponding situation option from among the possible is solved in the course of removing the uncertainty. Removing uncertainty enables to make informed decisions and take action. This is the guiding role of information.

The situation of maximum uncertainty presupposes presence of several equally probable alternatives (options), i.e. neither option is preferred. Moreover, the more equally probable options are observed, the greater the uncertainty, the more difficult it is to make an unambiguous choice and the more information is required for this to be obtained. For $N$ variants, this situation is described by the following probability distribution: $\left\{\frac{1}{N}, \frac{1}{N}, \cdots, \frac{1}{N}\right\}$.

The minimum uncertainty is $0$, i.e. this is a situation of complete certainty, meaning that the choice has been made and all the necessary information has been obtained. The probability distribution for a completely certain situation looks like this: $\{1, 0, \dots, 0\}$.

## 3. Methods to improve reliability

To improve the reliability of cloud systems, a mathematical apparatus, standardization, load balancing, protection from external influences and the choice of data storage schemes can be provided.

If the above methods did not give the desired result, then it is necessary to use a reservation [12-16].

Let's consider the factors affecting the survivability of cloud systems [17]. An important difference in the task of assessing survivability from other related tasks, for example, assessing reliability, is that, as a rule, it is impossible to use the concept of the probability of occurrence of certain situations. To increase survivability, one can apply such a number system in which the loss of some part would not lead to the termination of the functioning of the entire system at all.

That is why the role of number system choice for the functioning of the cloud storage system increases. The number system mainly determines the model of the system's reliability, the method of redundancy, the prevention of cascading failures and the growth of arising errors. One of the natural indicators of the qualitative measurement of survivability is the indicator $E$ preserved by the system after a fixed set of impacts.

For cloud storage systems, the main indicators of reliability are:

1. Probability of no-failure operation in the time interval from 0 to $t_0$

$$p(t_0) = p(0; t_0) = p\{\varepsilon \geq t_o] = 1 - F_1(t_0),$$

where $F_1(t) = p[\varepsilon_1 \leq t]$ is distribution of failures in time to the first failure.

2. The probability of failure-free operation of the facility in the time interval from $t$ to $t + t_0$

$$p(t, t + t_0) = p\{\varepsilon_1 \geq t + t_0 | \varepsilon_1 > t] = \frac{p(0, t + t_0)}{p(0, t)} = \frac{p(t + t_0)}{p(t)}$$

3. Density of failure distribution

$$f(t) = \frac{d}{dt}F(t) = -\frac{d}{dt}p(t)$$

4. The rate of failure of objects at the time

$$\mu(t) = \frac{1}{1 - F(t)} \cdot \frac{d}{dt} \cdot F(t)$$

The main practical method for improving the reliability and survivability of cloud storage systems is redundancy. Redundancy [18-21] is understood as a method of increasing the cloud system reliability by introducing additional clouds in excess of the minimum required for the normal functioning of the system.

Basic types of reservation:

1. Structural.
2. Temporary.
3. Functional.
4. Load.

The highest stability, reliability and survivability will be possessed by a system in which these methods are harmoniously combined and mutually penetrate each other [22, 23]. These are the characteristics of the Residue Number System (RNS). Using the RNS to build cloud storage systems provides with backup and working clouds (for each of the RNS bases), and also harmoniously use all types of redundancy described above.

## 4. Information reservation in cloud storage systems

The Residue Number System is a number system in which numbers are represented as a set of non-negative residues $x_1, x_2, \dots, x_n$ in coprime modulus $p_1, p_2, \dots, p_n$.

Let the numbers $X$ be represented by the residues $x_1, x_2, \dots, x_n$ at the bases $p_1, p_2, \dots, p_n$We will assume that $k$- residues are sufficient for an unambiguous representation of the number $X$, and $k < n$, $p_1 < p_2 < \dots < p_n$, while the working bases are $p_1, p_2, \dots, p_k$.

The range of unambiguous representation for the selected modulus is equal to the product of these modulus $P_n = \prod_{i=1}^{n} p_i$.

Thus, in the representation of the number $X$ the residues $x_1, x_2, \dots, x_n$ can be discarded any $r$-residues without compromising the uniqueness of the representation of the number $X$, as a result of which the RNS can control errors and it is a nonlinear code, which is called the $R$-code [24-27 ] [24-27].

In order to assess the ability of the RNS to control errors, we introduce the concept of the weight $W_{P_n}(|X|_{P_n}^{+})$ of the number $X$.

The weight $W_{P_n}\left(|X|_{P_n}^{+}\right)$ of the number $X$ will be considered equal to the number of nonzero residues. This definition of the weight of a number corresponds to the definition of the weight of the code in the Hamming metric.

In the symbol $W_{P_n}(|X|_{P_n}^{+})$ the subscript $P_n$ at $W$ shows that the number $X$ is represented by such a number of residues that $X \in |\cdot|_{P_n}^{+}$.

Obviously, the number of residues that represent the number $X$, in this case, is equal to $\omega$. The argument $|X|_{P_n}^{+}$ means that $X \in |\cdot|_{P_n}^{+}$.

With this definition of the number's weight $W_{P_n}(|X|_{P_n}^{+})$ it is possible to calculate it both directly from $X$ and from the totality of residuals $x_1, x_2, \dots, x_n$.

The concept of the weight of the number [28, 29] $X$, represented by the residues, can be used to define the concept of the distance between two points in space, each of which corresponds to the $X_1$ and $X_2$.

The distance $d_{X_1,X_2}$ between the points of space$X_1$ and $X_2$ is defined as the weight of the difference between $X_1$ and $X_2$.

$$d_{X_1,X_2} = W_{P_n}(|X_1 - X_2|_{P_n}^{+})$$

To determine the correcting capabilities of the code, we calculate the average weight $\overline{W}_{P_n}(|\cdot|_{P_n}^{+})$ of nonzero complexes. If the number $X$ is changed in the range$|\cdot|_{P_n}^{+}$, then the period for dividing the zeros is $p_i$, while the number of zeros in the base $p_i$ will be

$$\omega_i = \frac{P_k}{p_i} - 1, (i = \overline{1,k})$$

Then, respectively, the number of nonzero elements in the base is equal to

$$\overline{\omega_i} = P_k - 1 - \omega_i = P_k - \frac{P_k}{p_i}$$

The sum of the weights $S$ of the residues $x_i (i = 1,2,..,k)$, when $X$ is from zero to $P_k - 1$, is determined by the following expression:

$$S = \sum_{i=1}^{k} \left(P_k - \frac{P_k}{p_i}\right) = P_k \left(k - \sum_{i=1}^{k} \frac{1}{p_i}\right)$$

The average weight $\overline{W}_{P_n}(|\cdot|_{P_n}^{+})$ can be defined as

$$\overline{W}_{P_n}\left(|\cdot|_{P_n}^{+}\right) = \frac{S}{P_k - 1} = \frac{P_k}{P_k - 1} \cdot \left(k - \sum_{i=1}^{k} \frac{1}{p_i}\right)$$

The more the code is adapted to error correction, the more the numbers in the code representation differ from each other, i.e. the greater the code distance. Moreover, the distance will be different between different numbers [30-33].

If we determine the average weight of the numbers forming the zero space, then the minimum code distance will never exceed the upper bound of the minimum code distance. The minimum weight of elements of the zero-code space is

$$d_{min} = \min\left\{W_{P_n}\left(|\cdot|_{P_n}^{+}\right)\right\} = \omega - k + 1 \approx r + 1$$

With $d_{min} \approx r + 1$ we are guaranteed to detect any error. At the same time, the use of RNS allows detecting more errors. For example, an RNS with one redundant base, the value of which is greater than any of the working ones, allows detecting 100% of single errors and 95% of double errors.

It is also possible to detect all errors of multiplicity $t$ if $t < d_{min}$. This means that the RNS allows detecting errors of a given multiplicity and the multiplicity of detected errors is determined by the minimum code distance $d_{min}$.

Let the number $X = (x_1, x_2, .., x_n)$ be distorted, and instead of the number $X$ we have $\tilde{X} = (\widetilde{x_1}, \widetilde{x_2}, \dots, \widetilde{x_n})$, such that $\tilde{x} \in |\cdot|_{P_n}^+$. Let us calculate the difference $e$ between $\tilde{X}$ and $X$, which will determine as the magnitude of the error:

$$e = |\tilde{x} - x|_{P_n}^+$$

Since errors in different bit digits are independent of each other, then $e$ can be represented similarly to the representation of the number $X$, i.e. residuals $e = (e_1, e_2, \dots, e_n)$, which will be determined as $e_i = |\widetilde{x_i} - x_i|_{P_n}^+$.

Using the accepted residuals $\widetilde{x_1}, \widetilde{x_2}, \dots, \widetilde{x_n}$ we calculate the number $\tilde{X}$ by solving the comparison systems:

$$\tilde{X}_\iota \equiv \widetilde{x_\iota}(mod\ p_i), (i = 1,2, \dots, k)$$

To solve this system of comparisons, we use the method of orthonormal vectors of the form [25]

$$B_i = (0, 0, \dots, 1, \dots, 0) = \frac{m_i P_k}{p_i}$$

where $m_i \in |\cdot|_{p_i}^+$, such that $B_i \equiv 1(mod\ p_i)$.

With the help of orthonormal vectors, the number $X$ can be represented as

$$X = \sum_{i=1}^{k} x_i B_i - r_{P_k} x\left(|X|_{P_k}^+\right) P_k$$

where $r_{P_k}\left(|X|_{P_k}^+\right)$ is a function of the number rank, which for the value of the argument $|X|_{P_k}^+$ takes a value from calculating $X$ using orthonormal vectors $X \in |\cdot|_{P_k}^+$.

Having calculated the number $\tilde{X}$ we find the values of the residuals $\widetilde{x_1}, \widetilde{x_2}, \dots, \widetilde{x_n}$ of this number based on $p_{k+1}, \dots, p_n$.

Syndromic components are defined as

$$C_j = \left|x_{k+j} - \widetilde{x_{k+j}}\right|_{p_{k+j}}^+, j = 1, 2, \dots, n - k$$

Let us show that the fact of an error can be determined by the syndromic components.

We represent the number $X$ as

$$\tilde{X} = |X + e|_{P_k}^+$$

Suppose that only the residuals on the bases $p_1, p_2, \dots, p_k$, which are considered informational, are affected by errors. Taking this into account, we have

$$C_j = \left|\left|\sum_{\phi \in J_k} e_\phi B_\phi\right|_{p_k}^+\right|_{p_{k+j}}^+$$

where $e_\phi$ is an error value in the $\phi$-digit, $J_k$ is a set formed from the numbers of information bases, the residues of which are distorted.

Due to the fact that the quantity $e_k$ is not zero and that the excess grounds satisfy the condition of mutual simplicity, the inequality to zero of at least one syndromic component indicates the presence of errors.

Let's give an example of error detection. Let the number be represented in the RNS by the residues $x_1, x_2, \dots, x_n$, when converting the number to the positional number system (during data recovery) errors may occur. Therefore, after the data recovery $\tilde{X} = (\widetilde{x_1}, \widetilde{x_2}, \dots, \widetilde{x_n})$ is performed, we compare the number $\tilde{X}$ with the number $P_k$, and if $\tilde{X} \geq P_k$, then we conclude that an error occurred during the recovery.

Let's give another example. In the generalized positional number system (GPNS), the number $X \in |\cdot|_{p_n}^+$ can be represented in the form

$$X = \alpha_1 + \alpha_2 p_1 + \alpha_3 p_1 p_2 + \cdots + \alpha_n \prod_{i=1}^{n-1} p_i \qquad (1)$$

We will sequentially find the digit values $\alpha_i (i = 1,2, \dots, n)$, solving comparisons of the form $X \equiv x_i (mod\ p_i)$ starting with $i = 1$. Since all terms except for $\alpha_1$ are identically zero modulo $p_1$, hence $\alpha_1 = x_1$.

Solving $X \equiv x_i (mod\ p_2)$, $x_1 + \alpha_2 p_1 \equiv x_2 (mod\ p_2)$, therefore

$$\alpha_2 \equiv \frac{x_2 - x_1}{p_1} (mod\ p_1) \text{ or else } \alpha_2 = \left|\frac{x_2 - x_1}{p_1}\right|_{p_2}^{+}$$

Continuing the process of calculating the bit digits of the GPNS at the $k$-th step, we get

$$\alpha_k = \left|\frac{x_k - \sum_{i=1}^{k-1} \alpha_i \prod_{i=1}^{i} p_i}{\prod_{i=1}^{k-1} p_i}\right|_{p_k}^{+}, \qquad (k = 1, 2, \dots, n)$$

As a result of converting the number $X = (x_1, x_2, \dots, x_n)$ into the GPNS, we have the number $X = (\alpha_1, \alpha_2, \dots, \alpha_n)$.

If $X \in |\cdot|_{p_n}^{+}$, then $\alpha_{k+1}, \dots, \alpha_n$ will be equal to zero. This property can be used to receive data without error. The advantage of this algorithm is manifested in the fact that if our task is to establish the fact that errors have occurred, then if any $\alpha_{k+1}, \dots, \alpha_n$ is not equal to zero, the convertion process stops. This reduces the amount of computation required to detect errors.

## 5. Conclusions

This article proposes an approach to improving reliability and survivability based on modular arithmetic. The proposed approach makes it possible to increase the survivability of cloud storage systems, as well as to increase the reliability and fault tolerance of the data storage. To increase fault tolerance in the event of a failure, the redistribution of the processed data is applied. The introduction of low redundancy allows processing or restoring stored data in the event of a management server failure. This model allows recovering saved data in the event of a failure of one or more cloud servers. However, further research is needed to assess its efficiency in real systems. This will be the subject of our future work on a comprehensive experimental study of multipurpose optimization with real cloud providers.

**Acknowledgements** The reported study was funded by RFBR, project number 20-37-70023

## 6. References

[1] M. Babenko, N. Kucherov, A. Tchernykh, N. Chervyakov, E. Nepretimova, I. Vashchenko. Development of a Control System for Computations in BOINC with Homomorphic Encryption in Residue Number System, International Conference BOINC-Based High Performance Computing: Fundamental Research and Development, BOINC: FAST 2017: 77-84.

[2] R.L. Grossman, Y. Gu, M. Sabala, W. Zhang. Compute and storage clouds using wide area high performance networks, Future Generation Computer Systems (2009): 179-183.

[3] M.O. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance, Journal of the ACM (JACM)(1989): 335-348.

[4] A. Tchernykh, V. Miranda-Lopez, M. Babenko, F. Armenta-Cano, G. Radchenko, A.Y. Drozdov, A. Avetisyan. Performance evaluation of secret sharing schemes with data recovery in secured and reliable heterogeneous multi-cloud storage, Cluster Computing (2019): 1173-1185.

[5] A. Celesti, M. Fazio, M. Villari, A. Puliafito. Adding long-term availability, obfuscation, and encryption to multi-cloud storage systems, Journal of Network and Computer Applications. (2016): 208-218.

[6] A. Tchernykh, U. Schwiegelsohn, E. Talbi, M. Babenko. Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability, Journal of Computational Science (2019): 100581.

[7] H. Abu-Libdeh, L. Princehouse, H. Weatherspoon. RACS: a case for cloud storage diversity, Proceedings of the 1st ACM symposium on Cloud computing (2010): 229-240.

[8] K.D. Bowers, A. Juels, A. Oprea. HAIL: A high-availability and integrity layer for cloud storage, Proceedings of the 16th ACM conference on Computer and communications security (2009): 187-198.

[9] A.G. Dimakis, K. Ramchandran, Y. Wu, C. Suh. A survey on network codes for distributed storage, Proceedings of the IEEE (2011): 476-489.
[10] Z. Erkin, T. Veugen, T. Toft, R.L. Lagendijk. Generating private recommendations efficiently using homomorphic encryption and data packing, IEEE transactions on information forensics and security (2012): 1053-1066.
[11] M. Babenko, A. Tchernykh, N. Chervyakov, V. Kuchukov, V. Miranda-Lopez, R. Rivera-Rodriguez, Z. Du, E.G. Talbi. Positional Characteristics for Efficient Number Comparison over the Homomorphic Encryption, Programming and Computer Software (2019): 532-543.
[12] Z. Kong, S.A. Aly, E. Soljanin. Decentralized coding algorithms for distributed storage in wireless sensor networks, IEEE Journal on Selected Areas in Communications (2010): 261-267.
[13] M. Li, W. Lou, K. Ren. Data security and privacy in wireless body area networks, IEEE Wireless communications (2010): 51-58.
[14] H.Y. Lin, W.G. Tzeng. A secure erasure code-based cloud storage system with secure data forwarding, IEEE transactions on parallel and distributed systems (2012): 995-1003.
[15] L.J. Pang, Y.M. Wang. A new (t, n) multi-secret sharing scheme based on Shamir's secret sharing, Applied Mathematics and Computation (2005): 840-848.
[16] A. Parakh, S. Kak. Space efficient secret sharing for implicit data security, Information Sciences (2011): 335-341.
[17] A. Parakh, S. Kak. Online data storage using implicit security, Information Sciences (2009): 3323-3331.
[18] S. Ruj, A. Nayak, I. Stojmenovic. DACC: Distributed access control in clouds, 2011 International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11 (2011): 91-98.
[19] B.K. Samanthula, Y. Elmehdwi, G. Howser, S. Madria. A secure data sharing and query processing framework via federation of cloud computing, Information Systems (2015): 196-212.
[20] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A.G. Dimakis, R. Vadali, S. Chen, D. Borthakur. Xoring elephants: Novel erasure codes for big data, Proceedings of the VLDB Endowment (2013): 325-336.
[21] N.B. Shah, K.V. Rashmi, P.V. Kumar, K. Ramchandran. Interference alignment in regenerating codes for distributed storage: Necessity and code constructions, IEEE Transactions on Information Theory (2012): 2134-2158.
[22] C. Wang, Q. Wang, K. Ren, N. Cao, W. Lou. Toward secure and dependable storage services in cloud computing, IEEE transactions on Services Computing (2012): 220-232.
[23] J.J. Wylie, M.W. Bigrigg, J.D. Strunk, G.R. Ganger, H. Kiliccote, P.K. Khosla. Survivable information storage systems, Computer (2000): 61-68.
[24] C.C. Yang, T.Y. Chang, M.S. Hwang. A (t, n) multi-secret sharing scheme, Applied Mathematics and Computation (2004): 483-490.
[25] S.J. Lin, W.H. Chung, Y.S. Han. Novel polynomial basis and its application to reed-solomon erasure codes, IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS) (2014): 316-325.
[26] A. Tchernykh, M. Babenko, N. Chervyakov, V. Miranda-Lopez, V. Kuchukov, J.M. Cortes-Mendoza, M. Deryabin, N. Kucherov, G. Radchenko, A. Avetisyan. AC-RRNS: Anti-collusion secured data sharing scheme for cloud storage, International Journal of Approximate Reasoning (2018): 60-73.
[27] D.T. Liu, M.J. Franklin. GridDB: a data-centric overlay for scientific grids, Proceedings of the Thirtieth international conference on Very large data bases – VLDB Endowment (2004): 600-611.
[28] A. Tchernykh, M. Babenko, N. Chervyakov, J.M. Cortes-Mendoza, N. Kucherov, V. Miranda-Lopez, M. Deryabin, I. Dvoryaninova, G. Radchenko. Towards mitigating uncertainty of data security breaches and collusion in cloud computing, 28th International Workshop on Database and Expert Systems Applications (DEXA) (2017): 137-141.
[29] D. Amrhein, S. Quint. Cloud computing for the enterprise: Part 1: Capturing the cloud, DeveloperWorks, IBM (2009): 121-126.

[30] A. Tchernykh, M. Babenko, N. Chervyakov, V. Miranda-Lopez, A. Avetisyan, A.Yu. Drozdov, R. Rivera-Rodriguez, G. Radchenko, Z. Du. Scalable Data Storage Design for Non-Stationary IoT Environment with Adaptive Security and Reliability, IEEE Internet of Things Journal (2020).
[31] N. Chervyakov, M. Babenko, A. Tchernykh, N. Kucherov, V. Miranda-Lopez, J.M. Cortes-Mendoza. AR-RRNS: Configurable Reliable Distributed Data Storage Systems for Internet of Things to Ensure Security, Future Generation Computer Systems (2019): 1080-1092.
[32] M. Babenko, N. Chervyakov, A. Tchernykh, N. Kucherov, M. Shabalina, I. Vashchenko, G. Radchenko, D. Murga. Unfairness correction in P2P grids based on residue number system of a special form, 28th International Workshop on Database and Expert Systems Applications (DEXA) (2017): 147-151.
[33] I. Foster, C. Kesselman. The Grid 2: Blueprint for a future computing infrastructure, Elsevier, Waltham: Morgan Kaufmann Publishers (2004): 737.