Towards Mitigating Uncertainty of Data Security Breaches and Collusion in Cloud Computing

Andrei Tchernykh
CICESE Research Center
Ensenada, Mexico
chernykh@cicese.mx

Mikhail Babenko, Nikolay Chervyakov
North-Caucasus Federal University,
Stavropol, Russia
mgbabenko@ncfu.ru, k-fmf-primath@stavsu.ru

Jorge M. Cortés-Mendoza
CICESE Research Center
Ensenada, Mexico
jcortes@cicese.mx

Nikolay Kucherov
North-Caucasus Federal University, Stavropol, Russia
nkucherov@ncfu.ru

Vanessa Miranda-López
CICESE Research Center
Ensenada, Mexico
vanessa.vsphinx@gmail.com

Maxim Deryabin, Inna Dvoryaninova
North-Caucasus Federal University,
Stavropol, Russia
maderiabin@ncfu.ru, innadv99@mail.ru

Gleb Radchenko
South Ural State University,
Chelyabinsk, Russia
gleb.radchenko@susu.ru

***Abstract*— Cloud computing has become a part of people's lives. However, there are many unresolved problems with security of this technology. According to the assessment of international experts in the field of security, there are risks in the appearance of cloud collusion in uncertain conditions. To mitigate this type of uncertainty, and minimize data redundancy of encryption together with harms caused by cloud collusion, modified threshold Asmuth-Bloom and weighted Mignotte secret sharing schemes are used. We show that if the villains do know the secret parts, and/or do not know the secret key, they cannot recuperate the secret. If the attackers do not know the required number of secret parts but know the secret key, the probability that they obtain the secret depends the size of the machine word in bits that is less than $1/2^{(l-1)}$. We demonstrate that the proposed scheme ensures security under several types of attacks. We propose four approaches to select weights for secret sharing schemes to optimize the system behavior based on data access speed: pessimistic, balanced, and optimistic, and on speed per price ratio. We use the approximate method to improve the detection, localization and error correction accuracy under cloud parameters uncertainty.**

Keywords— Uncertainty, Collusion, Cloud Computing, Secret Sharing Schemes, Redundant Residue Number System

I. INTRODUCTION

Cloud computing emerges as a paradigm where computing infrastructures are virtualized as a shared pool of configurable resources (e.g., networks, servers, storage, applications, etc.) provided as a service on demand, in a pay-per-use manner. However, together with essential advantages, it has one serious obstacle that causes that many potential users do not use it intensively. There exist high risks of confidentiality, integrity, and availability associated with the loss of information, denial of access for a long time, information leakage, and collusion.

The occurrence of technical failures, data security breaches and collusion is difficult to predict. This type of uncertainty is one of the main problems in the design of the reliable cloud IT infrastructure.

According to the security report of Kaspersky [1], the key problem of modern Internet technologies is the DDoS attacks. To improve security and minimize the risks of data loss and corruption, various methods are used, like, data replication, secret sharing schemes, Redundant Residue Number System (RRNS), erasure codes, regenerating codes, homomorphic encryption, etc. [2-5]. Despite having been extensively studied for decades, many aspects of distributed cloud storage remain unexplored. The mechanisms of mitigating risks of collusion have not been adequately addressed in the scientific literature.

Information technologies have become common in people's lives. As a result, the volume of stored (transmitted) data is increased up to 1.1 exabytes (89 exabytes) in 2016. According to OpenFog forecasts [6], data volume will be increased up to 2.3 exabytes (194 exabytes) in 2020. To ensure the reliability and security of such a big data in clouds, it is necessary to develop new mechanisms for their storage and processing.

In this work, we study a reliable secret sharing scheme in RNS, and propose mechanisms to solve the problem of cloud collusion by the simultaneous use of the Mignotte weighted secret sharing scheme and Asmuth-Bloom secret sharing scheme.

II. RELATED WORKS

A. Big Data and Cloud Computing

Massive growth of secured data and analytics in Cloud computing creates a new trend that focuses less on collecting online data, open geographic data, data from open portals, public social and educational data, etc., and more on confidential business, medical care, health and social care, etc. information. Smart cities, smart factories, and relevant science domains are examples of emerging big data security.

Addressing big data is a challenging and time-demanding task that requires a large computational infrastructure to ensure successful data storing, processing and analysis.

Big data utilizes distributed cloud storage technology rather than local computer or electronic device storage. Considering volumes of stored data, the important criteria are security, reliability, redundancy and scalability [7, 8].

Cloud security issues are the limiting factors to the design of the cloud infrastructure for data storage and processing. In [9], the authors showed how collateral attacks on a virtual machine allows to access to confidential data. Efficient mechanisms for

2378-3915/17 $31.00 © 2017 IEEE
DOI 10.1109/DEXA.2017.44

monitoring the stored data [10], and verification of computation results [5] are necessary.

The use of distributed storage systems is associated with the risk of cloud collusion [11], which may occur because of unfaithful employees. Using the cloud collusion, an attacker can access confidential data in storage systems RACS [12], DepSky [13], RRNS [14], etc. Solutions to the cloud collusion problem are suggested in the works [10, 15]. Asymmetric ciphers allow to ensure the security of data storage, but cannot be applied to big data [10]. The HORNS RNS solution [15] allows to ensure data security, but is not effective in data processing.

According to the estimate of [16], if the Bigtable data are encrypted using the fully homomorphic cipher, then the Google query execution time would increase by about trillion times, which requires the optimization of modern encryption algorithms. One example of the effective construction of cloud storage is RRNS [5, 14].

B. Uncertainty in Cloud Computing

Cloud computing has considerable uncertainty on various levels of the computation, communication and storage. In [17], the authors showed that cloud elasticity has a positive impact on QoS, but adds a new uncertainty factor.

There are many types of uncertainties associated with cloud computing that should be considered in evaluation of the efficiency of provided service and impact on the performance, reliability and security. The cloud infrastructure assumes predictable and stable behavior of virtual machines and services in terms of performance. However, this assumption is not realistic. The actual performance depends on the underlying physical equipment, as well as the use of shared resources by other virtual machines assigned to the same host computers [2].

Uncertainty occurs in data processing, storage and transmission. The efficient use of the big data paradigm is directly related to solutions of the problem of multidimensional uncertainty.

Usually, in order to build a reliable data storage system, data replication is used. However, it leads to a dramatic increase in resource consumption.

Using communication awareness, [18] proposes a new model of CA-DAG cloud applications. CA-DAG eliminates the shortcomings of existing approaches and facilitates mitigating uncertainty. It allows to separate allocation of computational resources for tasks, and network resources for data transfer.

The scheduling problems are well studied. Many practical and theoretical solutions can be found. However, adaptive planning that allows to reduce uncertainty impact is rarely addressed [19]. This can lead to inefficient allocation of resources and poor energy consumption.

To handle uncertainties, the methods of probability theory, mathematical statistics, stochastic and fuzzy methods are widely used [17, 20]. In stochastic planning, the properties of problems are modeled as random variables due to the exact values are unknown until they are obtained. Online scheduling is characterized by a lack of knowledge about future. The decisions must be made each time as the tasks are released.

An alternative approach is discussed in [21]. The scheduling problem can be solved by using information about previously completed tasks, machine learning methods, regression, decision trees, etc. An important issue for the efficient implementation of such mechanisms is the length of the historical period. However, since characteristics of clouds and services are fast changing over the time, outdated historical information reduces their applicability.

III. Reliable and Secure Data Storage Scheme

Let consider the following scenario. User has confidential big data (secret) that cannot be stored in one cloud storage. He must to divide it on several pieces and store in different clouds. There are several threats.

Deliberate threats include unauthorized access to the information, interception, falsification, forgery, hacker attacks, etc. on data stored in one or several clouds. Cryptographic protocols and error correction codes can be used to reduce this risk even classical symmetric and asymmetric ciphers requires large computational power.

However, this approach is not sufficient for accidental threats that include errors, disasters, failures, etc. They could lead to the loss of one or several data pieces, inconsistency among different copies of the same data or inability to restore original data.

In this paper, we focus on collusion, when an attacker can access confidential data in one or several storages.

To reduce data redundancy, computational complexity of data encryption algorithms, and their low reliability in the case of collusion, we propose a cloud storage system based on the RNS homomorphic encryption. It provides a single method to ensure security, robustness, confidentiality, and encrypted data processing.

The secret is decomposed into a set of smaller encrypted parts, giving each cloud provider its own unique part. To reconstruct the secret, some of the parts or all of them are needed.

We consider three cases: If the villains know the secret parts, but do not know the secret key; If the attackers do not know the required number of secret parts and do not know the secret key, and If the attackers do not know the required number of secret parts and know the secret key.

A. Storage Model

Let i-th cloud provider has data access speed ch_i. Then each RNS module wieght can be represented using the formula $w_i = ch_i / \sum_{i=1}^{N} ch_i$.

Considering that characteristics of data access change over time, it is feasible to distribute the data load in such a way that i-th cloud has a set of n_i RNS modules. This approach allows to balance the data load under the condition of dynamic changing of technical parameters of cloud providers and to decrease the amount of transferred data.

Russian Federation President Grant SP-1215.2016, and Act 211 Government of the Russian Federation, contract № 02.A03.21.0011.

In order to decrease the computational complexity of arithmetic operations in RNS, it is feasible to use RNS modules of the size equals to the size of the machine word (l bits). Modules that equal to the power of two are not secure because they allow for a provider to know the piece of a confidential data equals to his data projection.

The volume of stored data in the i-th cloud is proportional to n_i. So, it is feasible to chose n_i equals to $n_i = \lceil 8 \cdot w_i \cdot L/l \rceil$, where L is the length of a block of coded data. Since the number of prime numbers in the range from 2 to x is roughly equal to $\pi(x) \approx x/\ln x$, the sufficient condition for the existence of RNS moduli set for storing the data according to the proposed scheme is

$$n = \sum_{i=1}^{N} n_i < \frac{l \cdot 2^{l-1} - 2^l}{l^2}.$$

Since the number of prime numbers of length 8-bits is 23, 16 bits – 3030, 24 bits – 513708, and 32 bits – 98182656, then, it is feasible to use obtained values in case $l \leq 32$ and in case $l > 32$ to use the estimation $2^{l-1}(l-2)/l^2$.

In RRNS, the computational security of the system depends on the parameters k, and n. Their appropriate values selection provides the necessary level of computational security and confidentiality. If $n_j \geq k$ exists, then j-th cloud provider can restore the encoded data and violate the confidentiality rules. On the other hand, the properties of RNS error correction codes allow to detect and correct errors that occur due to technical failures in data transfer and storage, or due to intentional data correction in collusion.

In [6], it is shown that the reliability of a system depends on r, where $r = n - k$. The bigger the value of r, the more reliable the system is, with growing redundancy.

Taking into account the volume of the data, the redundancy becomes the key factor in terms of distributed storage of big data. After the analysis of redundancy of such systems as RACS [12], DepSky [13], etc., it becomes clear that most suitable choice is $r < k$.

Given the technical characteristics and quality of service criteria, we offer four scenarios for data storage in the clouds:

- Pessimistic, when the minimum data access speed is used as weights for the weighted secret sharing scheme.
- Optimistic, when the maximum access speed to the data of each cloud provider is used to select the RNS moduli set;
- Balanced, when the average speed of access to the data obtained as a result of the tests is used to generate the system parameters;
- Balanced QoS, when the average speed per price of 1 GB is taken as a basis for generating the parameters of the data storage system.

To obtain the speed of data access and adapt to the dynamically changing technical characteristics of cloud providers, we store reports of the data access speed for each cloud provider for 30 days. Each day is described by five values: the minimum data access speed, maximum access speed, average speed, number of downloads, and number of refusals in data accessing. Considering these five characteristics and proposed scenarios, we determine parameters of the data storage system to ensure an appropriate level of security and reliability.

In this scenario, each cloud provider receives a chunk of data that consists of chunk identifier, chunk properties, projection of the original data, simplified digital signature, and moduli RNS. To compute the unique secret key, we use hash function based on SHA-3 algorithm [22].

B. Security

To study proposed secret sharing scheme, we use the concept of an asymptotically perfect Asmuth-Bloom scheme with zero knowledge [23].

Let the parameter p_0 be a secret key that allows to avoid cloud conspiracy [15]. In terms of arithmetic operations, the dynamic range of a system based on Asmuth-Bloom scheme is $[0, p_0)$, which is not suitable to construct the system of data security in cloud computing, due to the secret share is greater than the secret itself, and leads to more than n times increase in data volume.

Weighted Mignotte secret sharing scheme allows to optimally distribute the load among cloud storage providers in terms of data access and processing speed. However, it is not asymptotically perfect and ideal secret sharing scheme. If we use Mignotte scheme for cloud computing, the dynamic range is $[0, \alpha)$, which allows to increase the efficiency of the system and increase the number of arithmetic operations of multiplication k times, when compared to Asmuth-Bloom scheme.

To secure from cloud conspiracy under the condition of distribution of computational load among cloud storage providers, we combine the approaches of the two schemes: the Asmuth-Bloom and Mignotte.

Asmuth-Bloom and Mignotte schemes allow to implement adaptive schemes for data storage. To formalize the proposed scheme, we use the following notation. S - the message, $p_1, p_2, \ldots, p_n$ – set of pairwise coprime numbers (RNS moduli set), p_0 – an integer (adaptive parameter – key) that is coprime with each of $p_1, p_2, \ldots, p_n$. They satisfy the following four conditions.

1. $p_0 > S$
2. $\beta = \prod_{i=1}^{k} p_i > p_0 > \prod_{i=0}^{k-2} p_{n-i} = \alpha$
3. $2^{l-1} < p_1 < p_2 < \cdots < p_n < 2^l$
4. $\beta - \alpha > 2^{k \cdot (l-1).}$

Then, the shares of a secret are computed by the following formula:

$$\forall i \in [1, \ldots, n]: c_i = (S + r \cdot p_0) \bmod p_i$$

The modification of a condition 2, comparing to classic Asmuth-Bloom scheme, allows to increase the dynamic range of the system and number of multiplications $k-1$ times, and additions $2^{\left\lfloor \log_2 \prod_{i=0}^{k-3} p_{n-i} \right\rfloor}$ times. On the other hand, the

condition 2 allows to have the property of threshold secret sharing scheme. The weakening of the condition 1 allows to adapt the parameters for given level of security and reliability. The strengthening of condition 3 allows to ensure security similar to Rabin erasure codes [12, 13]. To prove this fact, we prove the following statement.

Statement 1. In proposed (k, n) secret sharing scheme, if an unauthorized coalition knows $k-1$ secret shares, and

- do not know the secret key p_0, then they can not know anything about the secret
- know the secret key p_0, then the probability that they find out the secret is less than $1/2^{(l-1)}$.

Proof. Without loss of generality, we can assume that there is a set of polynomial algorithms $p(\cdot)$, that allow for all $k_0 \geq k$ and any threshold value from $\left(s_{i_1}, s_{i_2}, .., s_{i_{k_0}}\right)$ to compute the values $C = S + r \cdot p_0$ (the value C can be computed using one of the following algorithm CRT, aCRT, nCRT, MRC [24] etc.).

For any set $I \subset \{1,2, \dots, n\}$, for which the cardinality of I is less than k, with polynomial algorithm $p(\cdot)$, it is possible to compute the value C^*, that satisfies the equality $C^* = |C|_{P_I}$, where $P_I = \prod_{i \in I} p_i$.

Taking into account condition 3, P_I satisfies the condition $P_I \leq \prod_{i=0}^{k-2} p_{n-i}$. Consequently, the probability to compute C with the known C^*, satisfies the equality

$$\Pr\big(p(I)\big) \leq \frac{1}{p_{n-k+1}} < \frac{1}{2^{l-1}}.$$

From condition 4, the cardinality of the set of all possible secret keys p_0 satisfies the condition

$$\prod_{i=1}^{k} p_i - \prod_{i=0}^{k-2} p_{n-i} = \beta - \alpha > 2^{k \cdot (l-1)}.$$

Consequently the probability to guess p_0 is less than $1/2^{k \cdot (l-1)}$. It means that the probability to compute the secret without knowing p_0 is less than $1/2^{(k+1) \cdot (l-1)}$. Under the condition that the cardinality of the set of all possible secrets is $[0, p_0)$, the intruders can not know anything about the secret. Δ

Corollary. In the proposed (k, n) scheme, a coalition with known k or more secret shares that does not know the secret key can know the secret with probability less than $1/2^{k \cdot (l-1)}$.

From the corollary, it follows that the proposed scheme allows to secure the data from the collusion.

IV. Experimental Analysis

The proposed system consists of several real Cloud Storage Providers, where we take into account available storage, and speed of data access. Table 1 shows the characteristics of storage/pricing and speed of access to data for the several cloud storage providers

To determine the speed of data access, we use a video file divided into chunks of 50, 150, 250, 350 and 450 MB. They are uploaded to all providers except Box, since Box supports files with maximum size of 250MB. Then, we downloaded them for three days each 4 hours. As a result, we obtained 30 measurements of corresponding speeds (Table 1). Testing is performed using of the Chrome browser version 58.0.3029.81 with Ethernet connection, which typically averages 143.71 Mbps download and 144.85 Mbps upload on Speedtest.net. Operating system is Windows 8.1 professional x64.

TABLE I. Speed of Access to Data, Storage and Pricing Plan for Cloud Storage Providers

Provider	Storage FREE	Storage	Price per month	Storage Business	Price per month	Low speed (MB/s)	High speed (MB/s)	Average speed (MB/s)		Average speed per price of 1 GB	
								Value	Rank	Value	Rank
Sync	5 GB	2TB	8.00$	1 TB	5.00$	1.61	4.41	2.78	9	695	2
DropBox	2 GB	2TB	12.50$	Unlimited	20.00$	2.50	8.92	4.93	5	788.8	1
IDrive	5 GB	1TB	5.34$	250 GB	6.22$	0.02	0.07	0.03	13	5.62	13
ElephantDrive	2 GB	1000GB	9.95$	2000 GB	39.95$	2.50	4.55	3.43	7	344.72	7
GoogleDrive	15 GB	100GB	1.99$	Unlimited	10.00$	4.55	12.50	7.87	3	395.48	6
MediaFire	10 GB	1TB	3.75$	100TB	40.00$	0.48	1.13	0.69	12	184	10
MEGA	50 GB	200GB	4.99€	4TB	29.99€	0.98	9.17	4.64	6	185.97	9
OneDrive	5 GB	1TB	6.00$	1TB	10.00$	1.67	4.78	2.44	10	406.67	5
Justcloud	1 GB	75GB	7.61$	1TB	10.69$	10.00	14.29	11.8	1	116.29	11
ICloud	5 GB	200GB	2.99$	2TB	19.99$	1.65	5.10	3.32	8	222.07	8
CloudMail	25 GB	512GB	7.80$	4TB	48.00$	5.49	10.71	7.89	2	517.91	3
YandexDisk	20 GB	100GB	1.30$	1TB	3.80$	3.80	9.78	6.21	4	477.69	4
Box	10 GB	100GB	4.00 €	Unlimited	12.00€	1.04	1.67	1.29	11	32.25	12

The CPU is Intel (R) Core (TM) i3-4330 CPU @ 3.5GHz, RAM 4 GB DDR3, HDD 1 TB.

We conclude that cloud storage providers can be classified in three groups based on quality of service per a price unit:

- The best quality for a price unit – Sync, DropBox.
- Good quality for a price unit – ElephantDrive, GoogleDrive, MediaFire, MEGA, OneDrive, ICloud, CloudMail, YandexDisk, justcloud.
- Satisfactory quality for a price unit – Box, IDrive.

We also can separate providers on three groups based on average data access speed:

- High access speed – justcloud, GoogleDrive, CloudMail, YandexDisk.
- Intermediate access speed – DropBox, Sync, ElephantDrive, MEGA, OneDrive, ICloud.
- Low access speed – IDrive, MediaFire, Box.

In experimental analysis, we use cloud storage providers included in 1st or 2nd group based on each criteria: justcloud, GoogleDrive, CloudMail, YandexDisk, DropBox, Sync, ElephantDrive, MEGA, OneDrive, and ICloud.

V. CONCLUSION

To minimize the harm caused by cloud collusion and data redundancy, we combine weighted Mignotte and threshold Asmuth-Bloom secret sharing schemes. To increase the efficiency of load balancing algorithms in conditions of uncertainty, we use RNS, which provides data security with reliability under collusion.

We propose four approaches to select weights for a weighted scheme for secret sharing. Three of them are based on data access speed: pessimistic, balanced, and optimistic, and on speed per price ratio. Since the algorithm of detection, localization and error correction is based on the comparison operation, we use the approximate method to compare numbers and increase its speed.

Provided theoretical analysis shows that our scheme improves the security and reliability under various collusion attacks. However, further study is required to assess its actual efficiency and effectiveness. This will be subject of future work providing a comprehensive experimental study of our four methods taking into account the improved decoding algorithm. It is important to provide multi-objective comparison with known approaches based on erasure codes, regeneration codes and secretion separation schemes.

REFERENCES

[1] "Kaspersky DDoS Intelligence Report for Q1 2016." URL: https://securelist.com/analysis/quarterly- malware-reports/74550 /kaspersky -ddos-intelligence-report-for-q1-2016/ 2016

[2] A. Tchernykh, U. Schwiegelsohn, E. Talbi, and M. Babenko, "Towards Understanding Uncertainty in Cloud Computing with risks of Confidentiality, Integrity, and Availability," J. Comput. Sci., 2016.

[3] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," Inf. Theory, IEEE Trans., vol. 56, no. 9, pp. 4539–4551, 2010.

[4] G. Ateniese, K. Fu, M. Green, & S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," ACM Trans. Inf. Syst. Secur., vol. 9, no. 1, pp. 1–30, 2006.

[5] N. Chervyakov, M. Babenko, A. Tchernykh, N. Kucherov, V. Miranda-López, J. Cortés-Mendoza, "AR-RRNS: Configurable Reliable Distributed Data Storage Systems for Internet of Things to Ensure Security," Futur. Gener. Comput. Syst., 2017

[6] "OpenFog Reference Architecture for Fog Computing," URL: https://www.openfogconsortium.org

[7] D. Hubbard and M. Sutton, "Top threats to cloud computing v1. 0," Cloud Secur. Alliance, 2010.

[8] A. Mora, Y. Chen, A. Fuchs, A. Lane, R. Lu, P. Manadhata, "Top ten big data security and privacy challenges," Cloud Secur. Alliance, 2012.

[9] T. Ristenpart, E. Tromer, H. Shacham, & S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," Proceedings of the 16th ACM conference on Computer and communications security, ACM, pp. 199-212, 2009.

[10] B. Samanthula, G. Howser, Y. Elmehdwi, & S. Madria, "An efficient and secure data sharing framework using homomorphic encryption in the cloud," Int. Work. Cloud Intell., pp. 1–8, 2012.

[11] M. Jensen, J. Schwenk, N. Gruschka, & L. L. Iacono, "On technical security issues in cloud computing," 2009 IEEE Int. Conf. Cloud Comput., no. 2009, pp. 109–116, 2009.

[12] H. Abu-Libdeh, L. Princehouse, & H. Weatherspoon, "RACS: a case for cloud storage diversity," SoCC, pp. 229–240, 2010.

[13] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "DepSky: dependable and secure storage in a cloud-of-clouds," ACM Trans. Storage, vol. 9, no. 4, p. 12, 2013.

[14] A. Celesti, M. Fazio, M. Villari, and A. Puliafito, "Adding long-term availability, obfuscation, and encryption to multi-cloud storage systems," J. Netw. Comput. Appl., vol. 59, pp. 208–218, 2016.

[15] M. Gomathisankaran, A. Tyagi, and K. Namuduri, "HORNS: A homomorphic encryption scheme for Cloud Computing using Residue Number System," in Information Sciences and Systems (CISS), 2011 45th Annual Conference on, 2011, pp. 1–5 .

[16] B. Schneier, "Homomorphic Encryption Breakthrough, Schneier on Security," July 9, 2009.

[17] A. Tchernykh, U. Schwiegelsohn, V. Alexandrov, E.-G. Talbi, "Towards Understanding Uncertainty in Cloud Computing Resource Provisioning," Procedia Computer Science, vol. 51, pp. 1772-1781, 2015.

[18] D. Kliazovich, J. Pecero, A. Tchernykh, P. Bouvry, S. Khan, A. Zomaya, "CA-DAG: Modeling Communication-Aware Applications for Scheduling in Cloud Computing Data Centers," IEEE 6th International Conference on Cloud Computing, pp. 277-284, 2013.

[19] A. Quezada-Pina, A. Tchernykh, J. Luis González-García, A. Hirales-Carbajal, J. Ramírez-Alcaraz, U. Schwiegelshohn, R. Yahyapour, V. Miranda-López. Adaptive parallel job scheduling with resource admissible allocation on two-level hierarchical grids. Future Generation Computer Systems 28/7 (2012) pp. 965-976. Elsevier

[20] I. Sotskov, F. Werner, "Sequencing and Scheduling with Inaccurate Data," Nova Science Pub, Applied Statistica Science, Minsk, 2014.

[21] S. Kianpisheh, S. Jalili, and N. Charkari, "Predicting Job Wait Time in Grid Environment by Applying Machine Learning Methods on Historical Information," Int. J. Grid Distrib. Comput., vol. 5, 3, 2012

[22] P. Pritzker, & P. Gallagher, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," Information Tech Laboratory National Institute of Standards and Technology, pp. 1–35, 2014.

[23] M. Quisquater, B. Preneel, & J. Vandewalle, "On the security of the threshold scheme based on the Chinese remainder theorem," Int. Workshop on Public Key Cryptography, pp. 199-210, 2002.

[24] P. Mohan, "RNS to binary conversion using diagonal function and Pirlo and impedovo monotonic function," Circuits, Systems, and Signal Processing, vol. 35, no. 3, pp. 1063-1076, 2016.