Analysis of Secured Distributed Cloud Data Storage Based on Multilevel RNS

Andrei Tchernykh[1,2]
CICESE Research Center,
Ensenada, BC, México
[1]chernykh@cicese.mx
South Ural State University,
Chelyabinsk, Russia
[2]chernykhan@susu.ru

Mikhail Babenko[3], Nikolay Chervyakov[4]
North-Caucasus Federal University
Stavropol, Russia
[3]mgbabenko@ncfu.ru,
[4]ncherviakov@ncfu.ru

Vanessa Miranda-López[5], Jorge M. Cortés-Mendoza[6]
CICESE Research Center
Ensenada, BC, México
[5]vanessamir.2813@gmail.com,
[6]jcortes@cicese.edu.mx

Zhihui Du[7]
Tsinghua University
Beijing, China
[7]duzh@tsinghua.edu.cn

Philippe OA Navaux[8]
Federal University of Rio Grande do Sul
Porto Alegre, Brazil
[8]navaux@inf.ufrgs.br

Arutyun Avetisyan[9]
Institute for System Programming of the RAS
Moscow, Russia
[9]arut@ispras.ru

***Abstract*— Cloud data storages are functioning in the presence of the risks of confidentiality, integrity, and availability related with the loss of information, denial of access for a long time, information leakage, conspiracy and technical failures. In this paper, we provide analysis of reliable, scalable, and confidential distributed data storage based on Multilevel Residue Number System (RNS) and Mignotte secret sharing scheme. We use real cloud providers and estimate characteristics such as the data redundancy, speed of data encoding, and decoding to cope with different user preferences. The analysis shows that the proposed storage scheme increases safety and reliability of traditional approaches and reduces data storage overheads by appropriate selection of RNS parameters.**

***Keywords*— multi-cloud; security; safety; reliability; Residue number system**

I. Introduction

Cloud computing becomes very important for enterprises IT infrastructures. However, there exist risks of safety, reliability, and availability of the data stored in clouds. Reliable storage systems use six main approaches: replication, secret sharing schemes, error correction codes, removal codes, regeneration codes, and homomorphic encryption [1].

Error correction codes based on Residue Number System (RNS) used for the design of distributed data storage system allow a high level of system reliability with the minimum data redundancy [2]. RNS and multi-level data flow is the perspective approach for reliable and confidential data storage [1].

Due to available risks of technical failures and safety of cloud services, we use the approach based on side-by-side storage of data in a set of the clouds. In Fig. 1, a distributed scheme in cloud-of-clouds is presented. Technical failure or data distortion appeared in the Cloud 3 is marked as ×. To build reliable and confidential data storage, the following approaches are used: Secret Sharing Schemes (SSS) [3, 4], Error Correction Codes in RNS (ECC-RNS) [2], Erasure Codes (EC) [5, 6], and Regeneration Codes (RC) [7]. The use of these approaches allows obtaining storable data in cases of a failure of one or several clouds.

II. Related Work

When the user stores data in a cloud, classic data replication possesses a high level of redundancy that requires repeated increase of the needed resources.

Google File System [8] and Bigtable technology uses a three-fold remark to achieve required reliability level. The system constructed on the base of Rabin's $(3, 5)$ Secret Sharing Schemes (SSS) requires 1.8 times less place keeping the same level of reliability. On the other hand, the classical replication possesses minimum expenses, when coding and decoding data unlike other approaches.

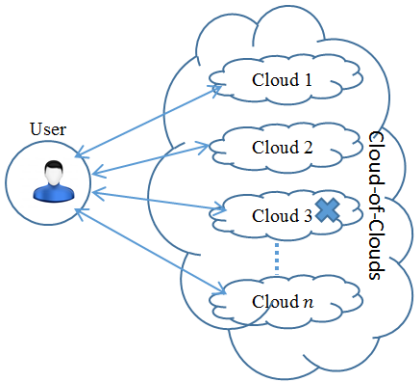

Fig. 1. Schema data storage cloud-of-clouds.

A. Data storage based on SSS

Threshold SSS allows to get access to data, if there are k or more data chunks. If SSS threshold is perfect and ideal, and there are less than k data chunks, the user will not obtain full information about the data.

The classical examples of such schemes are Shamir [9], Blakley [10], Asmuth-Bloom [11], Matrix Projections [12], etc.

However, ideal secret sharing schemes are not suitable for the cloud storage as they have the same redundancy as classical replication.

An alternative way is to use threshold computationally resistant SSS, such as SSS of Rabin [13], Mignotte [14] and others. It provides confidentiality of stored data with the minimal redundancy of data.

In cases of heterogeneity of technical characteristics of cloud storages, it is expedient to use weighted SSS that allows to distribute data between clouds according to their technical requirements. However, while using the weighted SSS, there can be a situation, when one cloud has all data. In this case, this scheme does not ensure data security.

B. Data storage based on ECC-RNS

The storage based on ECC-RNS provides high reliability and confidentiality level with the use of different cloud providers.

The BASE-64 coding adds 33% of redundancy, on average, comparing to the initial data size. Therefore, in the case of data storage circuit design, it is not expedient to use BASE-64.

The efficiency of data encoding algorithms in RRNS directly depends on a choice of RNS modules. Besides, in compliance with basic approaches to data decoding, RNS allows to receive the file even in case of a temporary or permanent failure of access to one or several cloud providers.

C. Data storage based on EC

Erasure Codes is a method of data protection in which a message is broken into k fragments or symbols, and expanded with m redundant symbols to provide protection from failures. As a result, resulting size of a message is $n = k + m$. In case of failure, EC generates another fragment instead of repairing it. Data processing is not possible with EC as it is not homomorphic. Lin et al. 2014 [15] show that EC has a complexity of $O(L \cdot \log_2 L)$, where L is the length of the code.

D. Data storage based on RC

Regeneration Codes is a technique to reconstruct a message from a partial corruption message. RC can recover a set of k coded fragments using an (n, k) Maximum Distance Separable (MDS) code approach. This reduces the total traffic required to repair the message [7]. However, RC does not allow high-performance computing, because it is not homomorphic [17]. To implement RC, it is needed a pseudorandom number generator with special properties [16].

III. Uncertainty in cloud computing

The uncertainty of technical failures has a serious impact on the reliability and security of data storage in the clouds [18]. Particular attention should be paid to the uncertainty arising from hacker attacks, DDoS attacks, and attacks on the synchronization key. The vulnerability of cloud systems to hacker attacks leads to an increase in the probability of breach of integrity and confidentiality of data.

To ensure the security and reliability of data storage in [19, 20], it is proposed to use AR-RRNS - configurable reliable distributed data storage systems. It is based on RNS and allows the reduction of the probability of data loss and increasing its security. To reduce the computational complexity of the data decoding algorithm, the AR-RRNS proposes an algorithm for computing the approximation of the rank of a number in RRNS. It allows a switch from the division with remainder to shift operation. Reduction in the size of the operands is achieved by the constraints used in Montgomery's modular multiplication algorithm.

To minimize the overhead under conditions of the uncertainty of the operation time of computing devices, an approach based on communication-aware directed acyclic graphs (CA-DAG) is proposed in [21-24]. In [21], it is shown that CA-DAG allows optimization of computational costs under conditions of uncertainty.

The study of the replication threshold value, depending on the available data, shows that replication allows minimization of the time delay, but leads to a great overhead increase.

As shown in [22], it is advisable to use classic data replication to minimize the damage resulting from the uncertainty of operations. Data replication greatly increases data redundancy, which leads to an increase in overhead for the maintenance of computing powers [19]. An alternative solution is RRNS error correction codes.

RRNS, due to its homomorphism, makes it possible to process encoded data. To increase the performance of the computer system under conditions of the uncertainty of technical failures, multilevel RNS can be used. Multi-level RNS provides reliable and secure long-term data storage in the clouds.

IV. Selection Of Multi-Level RNS Parameters

A. Multi-level RNS

The studies [25, 26] present the algorithm of modular computing in multi-level RNS based on the representation of the higher level bases in the form of lower level modules multiplication. The number R can be represented in a system of coprime numbers $p_1, p_2, \dots, p_n$, while a condition $\prod_{i=1}^{n} p_i > R$ is satisfied. The given system is the main system of the bases, which provide the possibility of operations in a given range $[0, R)$.

The maximum number, which can be received in this system is $(p_n - 1)^2$. Then, we can represent all digits of the main system in a new system with the bases $p_1^*, p_2^*, \dots, p_k^*$. In this case, the satisfaction of the following inequality is needed:

$$P^* = \prod_{i=1}^{k} p_i^* > (p_n - 1)^2$$

The maximum number, which can be received after multiplication, is $(p_k^* - 1)^2$. These last digits in the system of bases $p_1^*.p_2^*, \dots, p_k^*$ can be written in the system of bases $p_1^{**}.p_2^{**}, \dots, p_r^{**}$ if

$$P^{**} = \prod_{i=1}^{r} p_i^{**} > (p_k^* - 1)^2$$

Such a process of transfer to lower bases allows to change the representation of number R by hundreds of bits to numbers represented by 32 bits or by 64 bits, which can be easily processed on 32-bit and 64-bit processors.

In case of using multi-level RNS, the data processing is carried only at the top level, and the result of the computation is transformed to the most convenient form of output. The computation results are always correct if there is no overflow of number representation range both at the top level and at the bottom level.

B. The selection of multi-level RNS modules

Let us consider the modules selected from the main system of bases, the modules for each next level are chosen similarly. The same operating range can be realized using different sets of the bases. Sometimes, it is desirable that $P = \prod_{i=1}^{n} p_i$ is the biggest. For the maximum prime number, corresponding machine word is p_n , and maximum prime number that is less than p_n is p_{n-1}, etc. up to p_1.

In some cases, it is expedient to select one of the RNS modules; even it gives the opportunity to realize partition of the chosen range $[0, P)$ to equal sub-ranges P^+ и P^-, which will be used for representation of positive and negative numbers $P^+ = \left[0; \frac{P}{2} - 1\right]$ and $P^- = \left[\frac{P}{2}; P - 1\right]$. However, the existence of even base is not always convenient.

Increasing the range involves an increase in digit capacity of residues and in time expenditure at computing operations. One of the possible ways to decrease the values of bases is to construct the hierarchical residue number system.

To solve the problem of the RNS bases selection, it is necessary to define the optimization criterion of the set of bases. One of the main criteria is the condition of minimality of a number of binary digits, which are used in the design of the arithmetic devices. The criterion of minimality can be represented as

$$f(p_1, p_2, \dots, p_n) = \sum_{i=1}^{n} \lceil \log_2 p_i \rceil \rightarrow \min$$

Under a condition that modules of the main RNS satisfy an inequality:

$$P = \prod_{i=1}^{n} p_i \geq R$$

Let the modules of the main RNS satisfy a condition $2^{k_i - 1} \leq p_i < 2^{k_i}$, then

$$-n + \sum_{i=1}^{n} k_i \leq f(p_1, p_2, \dots, p_n) < \sum_{i=1}^{n} k_i$$

Considering that optimal module size from the point of view of computation is the size equal to the machine word size L, then, we will choose the value $n = \left\lceil \frac{\log_2 R}{L} \right\rceil$. The start parameters are: $k_1 = k_2 = \dots = k_n = L$. We compute the values $f(p_1, p_2, \dots, p_n)$, P, and $r = \left\lfloor \frac{P}{R} \right\rfloor$. Then $k_1 = k_2 = \dots = k_r = L - 1$.

The use of the given approach allows to design the balanced system from the point of view of the effectiveness of computation and data redundancy.

V. Simulation of Data Storage Scheme

Let us analyze the obtained characteristics for the balanced sets of modules of a special type: $\{2^n - 1, 2^n, 2^n + 1\}$, $\{2^n - 3, 2^n - 1, 2^n + 1, 2^n + 3\}$, $\left\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1\right\}$, $\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{\frac{n+1}{2}} + 1,\ 2^n + 2^{\frac{n+1}{2}} + 1, 2^{n-1} - 1\}$.

Considering that RNS modules satisfy the inequality $\prod_{i=1}^{k} p_i^* > (p_n - 1)^2$, we compute the minimum value of n_1 for each of modules sets. The results are presented in Table 1.

TABLE I. The exponent of N_1.

No	Moduli Set	n_1
1	$\{2^n - 1, 2^n, 2^n + 1\}$	$\left\lceil \frac{2n+1}{3} \right\rceil$
2	$\{2^n - 3, 2^n - 1, 2^n + 1, 2^n + 3\}$	$\left\lceil \frac{2n+1}{4} \right\rceil$
3	$\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{\frac{n+1}{2}} + 1,\ 2^n + 2^{\frac{n+1}{2}} + 1\}$	$\left\lceil \frac{2n+1}{5} \right\rceil$
4	$\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1$, $2^{n-1} - 1\}$	$\left\lceil \frac{n}{3} \right\rceil$

Using the results from Table 1, we compute the redundancy depending on the number of RNS layers. The data are presented in Fig. 2.

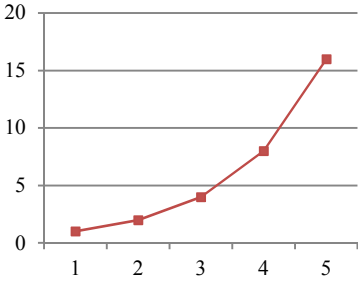

Fig. 2. The data redundancy depending on number of RNS layers

As we can see, the volume of stored data is 2^{L-1}, where L – the number of layers.

The dependence of one-megabyte data encoding time on the number of RNS layers and the chosen moduli set in milliseconds is shown in Fig. 3.

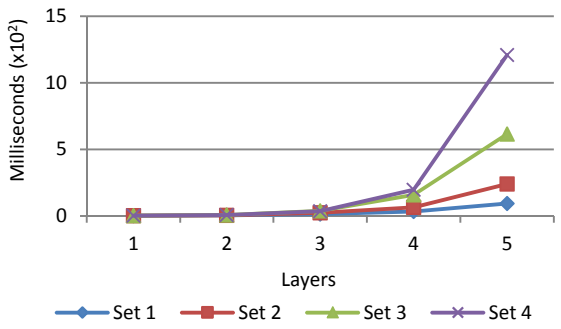

Fig. 3. Encoding time of 1 Mb.

The dependence of one-megabyte data decoding time on the number of RNS layers and the chosen moduli set in milliseconds is shown in Fig. 4.

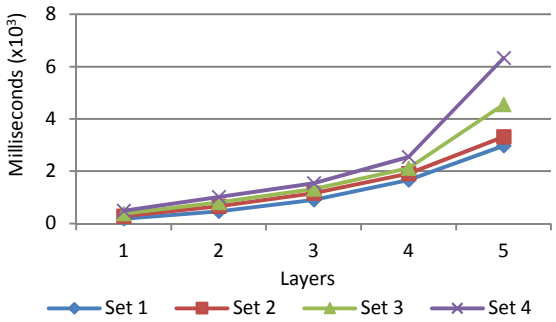

Fig. 4. Decoding time of 1 Mb.

IV. Conclusions

This paper provides an analysis of reliable, scalable, and confidential distributed data storage based on RNS. The analysis considers the redundancy, speed of data encoding, and decoding to cope with different user preferences. We show that the proposed storage increases safety and reliability of traditional approaches, and reduces an overhead of using data storage by appropriate selection of RNS parameters.

Acknowledgment

The work is partially supported by Russian Federation President Grant SP-1215.2016 and Russian Foundation for Basic Research (RFBR) 18-07-01224.

References

[1] 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*, 2016. DOI: 10.1016/j.jocs.2016.11.011

[2] 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*, vol. 59, pp. 208–218, 2016. DOI: 10.1016/j.jnca.2014.09.021

[3] V.Goyal, O. Pandey, A. Sahai, B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and Communications Security*, 2006, pp. 89-98. DOI: 10.1145/1180405.1180418

[4] G. Ateniese, K. Fu, M. Green, S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security (TISSEC)*, vol. 9, no. 1, pp. 1-30, 2006.DOI: 10.1145/1127345.1127346

[5] H. Abu-Libdeh, L. Princehouse, H. Weatherspoon, "RACS: a case for cloud storage diversity," in *Proceedings of the 1st ACM symposium on Cloud computing*, 2010, pp. 229-240. DOI: 10.1145/1807128.1807165

[6] A Bessani, M. Correia, B. Quaresma, F. André, P. Sousa, "DepSky: dependable and secure storage in a cloud-of-clouds," *ACM Transactions on Storage (TOS)*, vol. 9, no. 4, pp. 12, 2013. DOI: 10.1145/2535929

[7] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp.4539-4551, 2010. DOI: 10.1109/TIT.2010.2054295

[8] S. Ghemawat, H. Gobioff, S. T. Leung, "The Google file system," in *ACM SIGOPS operating systems review*, vol. 37, no. 5, 2003, pp. 29-43.

[9] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979. DOI: 10.1145/1165389.945450

[10] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the national computer conference,* 1979, vol. 48, pp. 313-317.

[11] C. Asmuth, J. Bloom, "A modular approach to key safeguarding," *IEEE Transactions on Information Theory*, vol. 30, no. 2, pp. 208-210, 1983. DOI: 10.1109/TIT.1983.1056651

[12] N. I. Chervyakov, M. G. Babenko, N. N. Kucherov, A. I. Garianina, "The effective neural network implementation of the secret sharing scheme with the use of matrix projections on FPGA," *Lecture Notes in Computer Science*, vol. 9142, pp. 3-10, 2015. DOI: 10.1007/978-3-319-20469-7_1

[13] T. Rabin, M. Ben-Or, "Verifiable secret sharing and multiparty protocols with honest majority," in *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, 1989, pp. 73-85. DOI: 10.1145/73007.73014

[14] M. Mignotte, "How to share a secret," in *Workshop on Cryptography*, 1982, pp. 371-375. DOI: 10.1007/3-540-39466-4_27

[15] S. J. Lin, W. H. Chung, Y. S. Han, "Novel polynomial basis and its application to reed-solomon erasure codes," in *IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS)*, 2014, pp. 316-325. DOI: 10.1109/FOCS.2014.41

[16] J. Liu, K. Huang, H. Rong, H. Wang, M. Xian, "Privacy-preserving public auditing for regenerating-code-based cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1513-1528, 2015. DOI: 10.1109/TIFS.2015.2416688

[17] H. C. Chen, P. P. Lee, "Enabling data integrity protection in regenerating-coding-based cloud storage: Theory and implementation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 407-416, 2014. DOI: 10.1109/TPDS.2013.164

[18] A. Tchernykh, M. Babenko, N. Chervyakov, J. M. Cortés-Mendoza, N. Kucherov, V. Miranda-López, M. Deryabin, I. Dvoryaninova, G. Radchenko, "Towards Mitigating Uncertainty of Data Security Breaches and Collusion in Cloud Computing," in *28th International Workshop on Database and Expert Systems Applications (DEXA)*, 2017, pp. 137-141. DOI: 10.1109/DEXA.2017.44

[19] N. Chervyakov, M. Babenko, A. Tchernykh, N. Kucherov, V. Miranda-López, J. M. Cortés-Mendoza, "AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security," *Future Generation Computer Systems*, 2017. DOI: 10.1016/j.future.2017.09.061

[20] 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," in *IEEE 28th International Workshop on Database and Expert Systems Applications (DEXA)*, 2017, pp. 147-151. DOI: 10.1109/DEXA.2017.46

[21] D. Kliazovich, J. E. Pecero, A. Tchernykh, P. Bouvry, S. U. Khan, A. Y. Zomaya, "CA-DAG: Communication-aware directed acyclic graphs for modeling cloud computing applications," in *CLOUD*, 2013, pp. 277-284. DOI: 10.1109/CLOUD.2013.40

[22] A. Tchernykh, J. E. Pecero, A. Barrondo, E. Schaeffer, "Adaptive energy efficient scheduling in Peer-to-Peer desktop grids," *Future Generation Computer Systems*, vol. 36, pp. 209-220, 2014. DOI: 10.1016/j.future.2013.07.011

[23] A. Hirales-Carbajal, A. Tchernykh, T. Röblitz, R. Yahyapour, "A grid simulation framework to study advance scheduling strategies for complex workflow applications," in *IPDPSW*, 2010, pp. 1-8. DOI: 10.1109/IPDPSW.2010.5470918

[24] D. Kliazovich, J. E. Pecero, A. Tchernykh, P. Bouvry,, S. U. Khan, A. Y. Zomaya, "CA-DAG: Modeling communication-aware applications

for scheduling in cloud computing," *Journal of Grid Computing*, vol. 14, no. 1, pp. 23-39, 2016. DOI: 10.1007/s10723-015-9337-8

[25] S. J Jassbi, M. Hosseinzadeh, K. Navi, "Redundant Multi-Level one-hot Residue Number System based error correction codes," *IEEE EWDTS*, 2010, pp. 491 - 494. DOI: 10.1109/EWDTS.2010.5742105

[26] N. Chervyakov, M. Babenko, A. Tchernykh, I. Dvoryaninova, N. Kucherov, "Towards reliable low cost distributed storage in multi-clouds," in *SIBCON*, 2017, pp. 1-6. DOI: 10.1109/SIBCON.2017.7998476