Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

AC-RRNS: Anti-Collusion Secured Data Sharing Scheme for Cloud Storage

Andrei Tchernykh[a,c,d,*], Mikhail Babenko[b], Nikolay Chervyakov[b], Vanessa Miranda-López[a], Viktor Kuchukov[b], Jorge M. Cortés-Mendoza[a], Maxim Deryabin[b], Nikolay Kucherov[b], Gleb Radchenko[c], Arutyun Avetisyan[d]

[a] *CICESE Research Center, Ensenada, BC, México*
[b] *North-Caucasus Federal University, Stavropol, Russia*
[c] *South Ural State University, Chelyabinsk, Russia*
[d] *Ivannikov Institute for System Programming RAS, Moscow, Russia*

ARTICLE INFO

Article history:
Received
Accepted

Keywords:
Uncertainty
Collusion
Multi-cloud
Cloud Computing
Secret Sharing Schemes
Residue Number System

ABSTRACT

Cloud security issues are important factors for data storage and processing. Apart from the existing security and reliability problems of traditional distributed computing, there are new security and reliability problems. They include attacks on a virtual machine, attacks on the synchronization keys, and so on. According to the assessment of international experts in the field of cloud security, there are risks of cloud collusion under uncertain conditions. To mitigate this type of uncertainty and reduce harms it can cause, we propose AC-RRNS algorithm based on modified threshold Asmuth-Bloom and Mignotte secret sharing schemes. We prove that the algorithm satisfies the formal definition of computational security. If the adversary coalition knows the secret shares, but does not know the secret key, the probability to obtain the secret is less than $1/(2^{l\cdot(k-1)}(2^{l-k}-1))$. The probability is less than $1/2^{(l-1)}$ with unknown secret shares and known secret key, and $1/2^{l\cdot k}$ with unknown secret key. Its complexity is equal to brute-force method. We demonstrate that the proposed scheme ensures security under several types of attacks. We propose approaches for selection of parameters for AC-RRNS secret sharing scheme to optimize the system behavior and data redundancy of encryption.

* A preliminary reduced version of this article appeared in Proceedings of UCC'17 - 1st International Workshop on Uncertainty in Cloud Computing, in conjunction with 28th International Conference on Database and Expert Systems Applications (DEXA'17) Lyon, France. August 28 - 31, 2017, p. 137-141, IEEE, 2017. 2378-3915/17, DOI: 10.1109/DEXA.2017.44
* Corresponding author: chernykh@cicese.mx

E-mail addresses: chernykh@cicese.mx (A. Tchernykh), mgbabenko@ncfu.ru (M. Babenko), ncherviakov@ncfu.ru (N. Chervyakov), vanessa.vsphinx@gmail.com (V. Miranda-López), vkuchukov@ncfu.ru (V. Kuchukov), jcortes@cicese.edu.mx (J. M. Cortés-Mendoza), maxim.deryabin@gmail.com {M. Deryabin}, nkucherov@ncfu.ru (N. Kucherov), gleb.radchenko@susu.ru (G. Radchenko), arut@ispras.ru (A. Avetisyan)

1. 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. 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, collusion and technical failures.

The occurrence of technical failures, data security breaches, and collusions is difficult to predict. These types of uncertainty is one of the main problems in the design of the reliable cloud IT infrastructure that is capable of mitigating their consequences. Also, there is a probability of denial of access to data and irretrievable data loss.

According to the Kaspersky security report (Kaspersky, 2016) [27], the key problem of modern Internet technologies is the DDoS attacks. To improve security and minimize the risks of data loss and corruption, the following approaches are used: data replication, secret sharing schemes (SSS), Redundant Residue Number System (RRNS), erasure codes, regenerating codes, homomorphic encryption, etc. (Tchernykh et al., 2016) [7], (Dimakis et al., 2010) [19], (Ateniese et al., 2006) [18], (Chervyakov et al., 2017) [33], (Tchernykh et al., 2018) [56]. Despite extensive study, many aspects of distributed cloud storage and mechanisms of mitigating risks of collusion have not been adequately addressed in the scientific literature.

Due to the development of Internet technologies, cloud providers permanently update, improve and modify data storage systems. Therefore, their technical characteristics are changing over the time: speed of data access, data storage capacity, tariff plans, the probability of failure, etc. It should be taken into account in systems design.

Encrypted data can be stored in the clouds, but cloud servers cannot perform calculations over encrypted data without first decrypting it when risks are also high. To protect information during the processing of confidential data using cryptographic methods, one should transfer keys to decrypt data only to trusted servers (Dimakis et al., 2010) [19], (Ateniese et al. 2006) [18].

Modern algorithms for symmetric and asymmetric encryptions are not suitable to provide data security in the cloud environment that needs data processing. Homomorphic encryption allows ensuring the confidentiality of the stored information and performing calculations over encrypted data without preliminary decoding, but with unacceptable time and resource consumption. Cloud computing is an evolving paradigm. It requires special approaches for security to mitigate risks of reliability, confidentiality, and integrity data.

To improve the reliability of data storage and transmission systems, error correction codes are widely used. They can be divided into three classes:

1. *Algebraic codes* are based on the polynomials of small degree over a finite field (also called Galois field). A large number of algebraic codes are known: Reed-Solomon codes, Bose–Chaudhuri–Hocquenghem (BCH) codes, algebraic-geometric codes on points of the elliptic curve, etc. The high speed of data encoding and decoding is the advantage of modern modifications of algebraic codes.
2. *Combinatorial codes* are based on the combinatorial properties of graphs. Sipser & Spielman 1996 [31], Zémor, 2001 [23] and Barg & Zemor, 2006 [1] study the properties and constructing combinatorial codes. Fu, Y., Sun, 2012 [47] study the speed of decoding of combinatorial codes and show that it is less than the decoding speed of algebraic codes. Combinatorial codes, as well as algebraic codes, do not have efficient algorithms for performing arithmetic operations over encoded data.
3. *Number-theoretic codes* are error-correction codes based on Residue Number System (RNS). For instance, a Mignotte secret sharing scheme and completely homomorphic cipher provide a proper level of reliability and security of data storage scheme. However, for effective application of RNS secret sharing schemes to processing big data, it is necessary to improve their security (Tchernykh et al., 2009) [57].

Information technologies have become common in people's life. As a result, the volume of stored/transmitted data is increased up to 1.1/89 exabytes in 2016. According to OpenFog forecasts (OpenFog) [36], data volume will be increased up to 2.3/194 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.

The application of RRNS to distributed storage and processing has great potential (Tchernykh et al., 2016) [7], (Chervyakov et al., 2017) [33]. It has the following advantages and limitations.

The advantages are:

- *Digits of the numbers* are formed independently. Therefore, each digit has the information about the whole data instead of a small part of the data. This property allows RRNS to detect, localize, correct errors, and act as a secret sharing scheme. It ensures reliable data storage under the conditions of technical failures and uncertainty of cloud parameters.
- *Reduced size* of the codeword as an element of an error-correcting code allows storing big volumes of data with efficient processing and verification of the result (Chervyakov et al., 2017) [33], (Miranda-López et al., 2017) [58] under the conditions of the uncertainty of technical failures.

The main shortcomings of RRNS include:

- *Computational complexity* of magnitude comparison of numbers, since the representation of a number does not give information about the magnitude of the number;
- *Euclidean division*. A comparison operation is required;
- *Overflow detection.* The simple criteria for dynamic overflow detection are not exists.

In this paper, we propose AC-RRNS a computational secure and reliable secret sharing scheme (SSS) in RRNS and study its security features. We solve the problem of cloud collusion by the simultaneous use of the ideas behind the Mignotte secret sharing scheme and asymptotically ideal Asmuth-Bloom secret sharing scheme.

By changing the requirements of the secret key, we increase the dynamic range of the system and reduce data redundancy comparing with the Asmut-Bloom scheme.

RNS has emerged as a key player in this endeavor. It is based on a puzzle introduced by the Chinese mathematician Sun-Tzu, later named as Chinese Remainder Theorem (CRT). Based on CRT, (Szabo & Tanaka, 1967) [35] invented RNS. RNS is a non-weighted data representation system that allows representing an integer number as a set of smaller numbers.

Initially, the main application area of RNS was Digital Signal Processing. In modern cryptosystem design, the main RNS application is related to Montgomery modular multiplication. Recently, the demand for parallel and low-power computations has led to the adaptation of RNS in emerging applications such as wireless sensor networks, cloud computing, distributed memory, and hybrid memory.

Because traditional encryption techniques and security protocols are not sufficient to protect data transmission and storage in the Cloud, researchers have turned their attention to alternative systems to further boost up cryptosystem performance. An alternative approach of the homomorphic encryption is a homomorphic cipher based on RNS. To apply RNS for encryption, the issues of confidentiality, integrity, and collusion need to be addressed (Chang et al., 2015) [11].

2. Related work

The fast-growing data and analytics in Cloud computing creates a new trend that focuses less on collecting online data, open geographic information, 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 security trends of big data.

Addressing big data is a challenging and time-demanding task that requires a large computational infrastructure to ensure successful data processing, analysis, and storing (Kirthica & Sridhar, 2018) [54], (Tchernykh et al., 2018) [59], (Massobrio et al., 2018) [60]. Big data utilizes distributed cloud storage technology rather than local computer or electronic device storages. Considering volumes of stored data, the important criteria are security, reliability, redundancy, and scalability (Hubbard & Sutton, 2010) [15], (Mora et al., 2012) [8].

Cloud security issues are important factors to design the cloud infrastructure for data storage and processing. In (Ristenpart et al., 2009) [45], the authors showed how collateral attacks on a virtual machine open access to confidential data. Efficient mechanisms for monitoring the stored data (Samanthula et al., 2012) [9] and verification of computation results (Chervyakov, 2017) [33] are necessary.

The use of distributed storage systems is associated with the risk of cloud collusion (Jensen et al., 2009) [30], which may occur because of unfaithful employees. Using the cloud collusion, an adversary can access confidential data (Abu-Libdeh et al., 2010) [24], DepSky (Bessani et al., 2013) [2], RRNS (Celesti et al., 2016) [3] storage systems, etc.

Solutions to the cloud collusion problem are suggested in the works (Samanthula et al., 2012) [9], (Gomathisankaran et al., 2011) [28]. Asymmetric ciphers allow to ensure the security of data storage, but cannot be applied to big data (Samanthula et al., 2012) [9]. The HORNS RNS solution (Gomathisankaran et al., 2011) [28] ensures data security but is not effective in data processing. According to (Schneier, 2009) [10], if the Bigtable data are encrypted using the fully homomorphic cipher, then the Google query execution time would increase by about trillion times. It requires the optimization of modern encryption algorithms. One example of the effective construction of cloud storage is RRNS (Chervyakov et al., 2017) [33], (Celesti et al., 2016) [3].

Cloud computing has considerable uncertainty on various levels of the computation, communication, and storage. There are many types of uncertainties associated with cloud computing that should be considered in an evaluation of the efficiency of provided service and impact on the performance, reliability, and security (Tchernykh et al., 2015) [7]. The cloud infrastructure assumes the predictable and stable behavior of virtual machines and services regarding 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 (Tchernykh et al., 2016) [7]. The efficient use of the big data paradigm is directly related to mitigating multidimensional uncertainty (Omri et al., 2018) [55].

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

Kliazovich et al., 2016 [16] propose a new model of CA-DAG cloud applications that use communication awareness. CA-DAG eliminates the shortcomings of existing approaches and facilitates mitigating uncertainty. It allows separating 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 reducing uncertainty impact is rarely addressed (Quezada-Pina et al., 2012) [5], (Tchernykh et al., 2009) [57]. The uncertainty can lead to inefficient allocation of resources and increased energy consumption.

The methods of probability theory, mathematical statistics, stochastic and fuzzy methods are widely used to handle uncertainties (Tchernykh et al., 2015) [7], (Sotskov & Werner, 2014) [50]. 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.

Kianpisheh et al., 2012 [43] discuss the scheduling problem that 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.

3. Redundant residue number system and cloud computing

The concept of cloud computing provides dynamic allocation of resources. Therefore, apart from the existing security and reliability problems of GRID computing, there are new security and reliability problems (Jensen et al., 2009) [29], (Subashini & Kavitha, 2011) [44]. They include the collusions (Gomathisankaran et al., 2011) [28], attacks on a virtual machine (Hubbard & Sutton, 2010) [15], attacks on the synchronization keys (Xiao & Xiao, 2013) [51], and so on. To reduce the uncertainty and minimize the risks of data security breaches and denial of access to data, it is reasonable to use error localization and correction codes of RRNS.

3.1 Redundant Residue Number System (RRNS)

In this system, the original number is represented as residues concerning a moduli set. The number is split into several smaller numbers, which are independent. Their processing can be done independently and concurrently. It makes the computations simpler and much faster.

Let p_i for all $i = \overline{1,n}$ be pairwise coprime numbers used as the module i of RRNS, p_0 - the secret key, and $n = k + r$. Then RRNS range P is defined as $P = \prod_{i=1}^{k} p_i$.

A secret (original data) can be represented as an integer number S, where $S \in [0, P)$. S is defined in RRNS as a tuple $S \xrightarrow{RRNS} (s_1, s_2, \dots, s_n)$, where $s_i = |S|_{p_i}$ is a share that represents the remainder of a division of S by p_i. In RRNS settings (k, n), using data from any k shares from n, we can recover $r = n - k$ data.

According to RRNS property, if the number of control modules is r, then the system can detect r and correct $\lfloor r/2 \rfloor$ errors. Redundancy of residues allows building a reliable data processing system with multiple error detection and correction (Chessa et al., 2004) [41], (Celesti et al. 2016) [3], (Chervyakov et al., 2017) [33], (Tchernykh et al., 2017) [6].

One special property of RRNS is the possibility to perform addition, multiplication, and subtraction in parallel and independently for each share (Szabo & Tanaka, 1967) [35]. Arithmetic operations are performed without carries among x_i, which, on the one hand, allows parallel processing of data in the clouds, and, on the other, provide confidentiality.

The use of control moduli in the RRNS provides reliability in the long-term storage of data and enables verification of the result (Celesti et al., 2016) [3]. To detect and correct errors in RRNS, projections method is used, which is equivalent to conversion from RNS to binary. However, the number of projections that need to be computed grows exponentially. The computation of one projection is already a computationally complex algorithm.

To optimize the algorithm, (Chervyakov et al., 2017) [33] proposed new algorithm of error detection and localization in RNS based on the approximate rank of a number and characteristic function. This method reduces the computational complexity of the decoding algorithm but requires larger memory volume to store the characteristic function in a table form.

If we consider that RRNS is not only the error detection, localization and correction code but also Mignotte secret sharing scheme, then we can conclude that RRNS can be used to ensure data security.

Gomathisankaran et al., 2011 [28] show that the Mignotte secret sharing scheme is a completely homomorphic cipher that allows processing data in an encrypted form and.

To solve the problem of cloud collusion, we propose AC-RRNS based on modification of Asmooth-Bloom and Mignotte schemes. The advantage of the proposed scheme is reducing the data redundancy in comparison with Asmooth-Bloom scheme

and ensuring of the computational security (see Section 7.1). Our solution allows to ensure the data security, unlike Mignotte scheme, with maintaining the values of the redundancy of data on the same level (see Section 7.2).

In the proposed solution, to ensure the reliability of data storage, the error correction properties of codes based on RRNS are used. Most scientific works on RRNS consider detection and correction only one error due to high reliability of modern computer technology. However, this approach is inapplicable to distributed storage systems, especially for big data, since the probability of the failure of several computing nodes is higher. Therefore, reliable storage systems require an error correction mechanism of RRNS. An efficient mechanism for detection, localization, and correction of errors in RRNS based on of an approximate value of a rank of a number is proposed in Chervyakov et al., 2017 [33].

3.2. RRNS properties

An unsigned integer S within range P can be represented using residue, computed by taking the least positive number of the division of S by p_i. To represent a signed integer S within P, P is divided into two sub-range (Chang et al., 2015) [11]. The lower half and upper half ranges are used to represent positive and negative integers, respectively (Szabo & Tanaka, 1967) [35]. Representation of numbers in RNS allows replacing operations over large numbers with operations over small numbers that are processed in parallel and independently.

For efficient implementation, it is important to choose the moduli set. There are several approaches to select them. For example, (Patronik & Piestrak, 2014) [38] study the moduli set of a special form $\{2^n - 1, 2^n, 2^n + 1\}$. They allow developing efficient algorithms such as conversion from binary number system to RNS and back, and arithmetic operations. However, they do not let to scale the system efficiently and require additional resources. Arithmetic operations for each module are designed individually, and aggregation of a new module requires optimization of RNS-to-binary conversion.

Phatak & Houston, 2016 [17] presented an approach for high-performance computing in RNS for GPU that uses prime numbers of a specific form suited well for big data storage and processing.

Well-known algorithms for RNS to binary conversion include mixed-radix conversion (MRC), Chinese remainder theorem (CRT), mixed-radix CRT (Bi & Gross, 2008) [40], new Chinese remainder theorems (nCRT) (Wang, 2000) [49], and their modifications.

The high computational complexity of the conversion is the reason why the researchers search for the methods to approximate the result. Some of them are: Chinese remainder theorems (aCRT) (Van Vu, 1985) [46], (Chervyakov et al., 2017) [34], core function (Burgess, 2003) [32], quotient function (Dimauro et al., 2003) [21], diagonal function (Dimauro et al., 1993) [20] (Mohan et al., 2016) [37], monic function Pirlo and Impedovo (Pirlo & Impedovo, 2013) [22].

4. Reliable and Secure Data Storage Scheme

Let consider the following scenario. The user has confidential data (secret) and decides not to store it in single cloud storage. He divides it into several chunks and stores them in different clouds.

There are several security threats.

- *Deliberate threats* include unauthorized access to the information, interception, falsification, forgery, hacker attacks, etc. on one or several clouds.
- *Accidental threats* 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.
- *Collusion threads* is an illegal agreement between two or more adversaries to gaining the full access to the private data.

Cryptographic protocols and error correction codes can be used to reduce deliberate threats but not sufficient for accidental threats.

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

There are three main groups of methods used to solve the collusion problem.

1. *Collusion detection.* To detect collusion, mechanisms based on fingerprinted data can be used (Boneh & Shaw, 1998) [14], (Ye et al., 2016) [12], (Ekodeck & Ndoundam, 2016) [42], etc. Fingerprinted data is a class of methods designed to put labels on data to detect users to whom the data was provided (Li et al., 2005) [48]. The objective is to detect an adversary who participates in collusion.
2. *Collusion prevention.* The basic idea of these methods is in denial of the data access to adversaries based on key distribution (Hur, 2013) [26], group managers (Zhu & Jiang, 2016) [52], access structure (Zhu & Jiang, 2016) [52], etc. The weak point of these methods is the centers of key distribution and group managers. In case of collusion with one of them, the adversary gets the full access to the data.

3. *Data security in case of collusion.* If adversaries participate in collusion, then a method of adding noise to the confidential data can be used. To add noise two main methods are used: based on a secret key and hidden RRNS modules (Gomathisankaran et al., 2011) [28], (Fernandes et al., 2014) [13]. Gomathisankaran et al., 2011 [28] proposed a modification of Asmooth-Bloom and Mignotte secret sharing schemes. However, as it is shown in Krawczyk, 1993 [25], Asmooth-Bloom scheme is not applicable in practice because of great data redundancy. In Section 5, we show that the Mignotte scheme does not solve the problem of cloud collusion.

Let us consider the methods of the second and third groups.

To prevent cloud collusion, Hur, 2013 [26] proposed a scheme based on secret key attributes, which allow decreasing the number of open keys. The author uses distributed key escrow composed of two parts: key generation center and storing center.

To solve the problem of collusion, Zhu & Jiang, 2016 [52] proposed a scheme based on the access structures and key distribution protocol. It is based on the use of bilinear pairing on the points of elliptic curves and a Diffie-Hellman discrete logarithm. The group manager uses the secret master key. In case of technical failures resulting in the loss or distortion of the secret master key, the operation of the entire system is disrupted. If there is a leak, then the adversary can gain complete control over the distributed data storage system.

Hur, 2013 [26], Zhu & Jiang, 2016 [52] considered operations over elliptic curve points based on bilinear pairing. Due to the master key security directly depends on the computational complexity of solving of the Diffie-Hellman discrete logarithm problem, the proposed efficient algorithm is not computationally secure.

An alternative solution to the problem of collusion is the use of secret sharing schemes based on RRNS, which allows the security and reliability of data during long-term storage and processing.

Asmooth-Bloom and Mignotte RRNS schemes ensure reliable data storage. The main drawback of the Asmooth-Bloom scheme is the large data redundancy. The Mignotte scheme has a low level of data security. To increase the security level of Mignotte, Gomathisankaran, et al., 2011 [28] use the RRNS moduli set as a secret key. However, it leads to increased redundancy.

Computationally secure schemes allow the security of stored data and reduction of data redundancy and network load by k times with respect to Asmooth-Bloom, Shamir, etc. (Krawczyk, 1993) [25].

In this paper, we propose AC-RRNS homomorphic encryption and computationally secure scheme based on RRNS and secret sharing scheme. It does not allow the adversary of data storage system to map original data (secret) to corresponding shares. It also prevents known-plaintext attacks, where the adversary knows all the data apart from the secret key and aims at discovering it. 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 that saved in memories of different providers.

We consider three cases:

1. Adversaries know the secret parts, but do not know the secret key;
2. Adversaries do not know the required number of secret parts and do not know the secret key,
3. Adversaries do not know the required number of secret parts and know the secret key.

Gomathisankaran et al., 2011 [28] proposed a modified Mignotte SSS approach named HORNS to collusion problem based on hidden RRNS modules.

In Section 5, we show that HORNS is less secured than Asmooth-Bloom scheme. In Section 6, we prove that our AC-RRNS is computationally secure.

5. Known-Plaintext attack of SSS based on RRNS

5.1. Known-Plaintext attack of HORNS

In HORNS, the set of modules RNS is used as a secret key. Due to the key is unknown, results of arithmetic operations performed on stored shares are not transformed to residual. It increases the size of stored data.

If after the attack, the secret and residuals become known, an adversary can identify the RRNS moduli by appropriate selection of possible moduli.

The basic idea of the method to compute hidden RRNS modules is the following. There are three cases of the module value for arbitrary integer t.

Case 1: if $2^t < p_i$, then $|2^t|_{p_i} = 2^t$.
Case 2: if $2^{t-1} < p_i \leq 2^t$, then $2^t = p_i + |2^t|_{p_i}$.
Case 3: if $2^t \geq p_i$, then $|2^t|_{p_i} \neq 2^t$.

The lower bound of t is more or equals to 1. To find an upper bound of t, we use Case 3, where $2^t \geq p_i$.

Let us take t from geometric series with the common ratio 2: $1, 2, 4, 8, 16, 32 \ldots$. Now, we estimate the lower and upper bounds more precisely: $t \in (2^{a-1}, 2^a]$, where $a = \lceil \log_2 \log_2 p_n \rceil$.

We can refine t by applying Case 2 and binary search. To illustrate the method, let us consider the following example.

Example 1. Let the parameters of HORNS secret sharing schemes are $k = 2$, $n = 4$ and RRNS modules are: $p_1 = 3221225473$, $p_2 = 3221225479$, $p_3 = 3221225533$, $p_4 = 3221225549$, where $\{p_1, p_2, p_3, p_4\}$ is the secret key.

Table 1 shows values of the parameter a.

Table 1
Selection of the parameter a

a	$S = 2^{2^a}$	$\lvert S\rvert_{p_1}$	$\lvert S\rvert_{p_2}$	$\lvert S\rvert_{p_3}$	$\lvert S\rvert_{p_4}$
0	$2^{2^0} = 2$	2	2	2	2
1	$2^{2^1} = 4$	4	4	4	4
2	$2^{2^2} = 16$	16	16	16	16
3	$2^{2^3} = 256$	256	256	256	256
4	$2^{2^4} = 65536$	65536	65536	65536	65536
5	$2^{2^5} = 4294967296$	1073741823	1073741817	1073741763	1073741747

Since for $a = 5$ the condition $S \neq \lvert S\rvert_{p_4}$ is satisfied, then $t \in (2^4, 2^5]$.

Now, we can estimate t more precisely (Table 2).

Table2
Refinement of the parameter t.

a	b	$t = \left\lceil \frac{a+b}{2} \right\rceil$	$S = 2^t$	$\lvert S\rvert_{p_1}$	$\lvert S\rvert_{p_2}$	$\lvert S\rvert_{p_3}$	$\lvert S\rvert_{p_4}$
16	32	24	$2^{24} = 16777216$	16777216	16777216	16777216	16777216
24	32	28	$2^{28} = 268435456$	268435456	268435456	268435456	268435456
28	32	30	$2^{30} = 1073741824$	1073741824	1073741824	1073741824	1073741824
30	32	31	$2^{31} = 2147483648$	2147483648	2147483648	2147483648	2147483648
31	32	32	$2^{32} = 4294967296$	1073741823	1073741817	1073741763	1073741747

From the results of Table 2 and Case 2, we can conclude that $t = 32$ and the secret key (RRNS modules) are:

$$p_1 = 2^{32} - \lvert S\rvert_{p_1} = 2^{32} - 1073741823 = 3221225473,$$
$$p_2 = 2^{32} - \lvert S\rvert_{p_2} = 2^{32} - 1073741817 = 3221225479,$$
$$p_3 = 2^{32} - \lvert S\rvert_{p_3} = 2^{32} - 1073741763 = 3221225533,$$
$$p_4 = 2^{32} - \lvert S\rvert_{p_4} = 2^{32} - 1073741747 = 3221225549.$$

Example 1 shows that the approach from Gomathisankaran et al., 2011[28] is not computationally secure and the secret key can be computed. As a result, Known-Plaintext attack enables to retrieve the confidential data of the user in the hidden RRNS modules framework.

The HORNS scheme is not computationally secure since it allows unambiguous mapping of original data to the shares.

5.2. Known-Plaintext attack of Asmooth-Bloom

In this section, we study Known-Plaintext attack on Asmooth-Bloom scheme. Similar to the Known-Plaintext attack of Asmooth-Bloom, we assume that the adversary knows several data sets (several secrets). Each set j includes original data $S^{(j)}$, RRNS moduli set, and RRNS shares: $C^{(j)} \xrightarrow{RRNS} (c_1^{(j)}, c_2^{(j)}, \ldots, c_n^{(j)})$. The shares are generated according to the following formula:

$$c_i^{(j)} = \left| S^{(j)} + p_0 \cdot rand_j \right|_{p_i}, \quad (1)$$

where $rand_j$ is the random number, p_0 is the secret key, and $\lvert x\rvert_p$ is the least nonnegative residue of x modulo p. To estimate the number of pairs $(S^{(j)}, C^{(j)})$ needed to recover the secret, we prove the following theorem.

Theorem 1. *To compute the secret key of Asmooth-Bloom scheme with a Known-Plaintext attack, it is necessary that the supremum value of j be equal to* $\lceil k \cdot log_2 p_0 \rceil$.

Proof. From Eq. (1), we obtain $c_i^{(j)} = S^{(j)} + p_0 \cdot rand_j - \alpha \cdot p_i$, where $\alpha \in Z$. Hence,

$$p_0 \cdot rand_j = c_i^{(j)} - S^{(j)} + \alpha \cdot p_i.$$
$$\left|p_0 \cdot rand_j\right|_{p_i} = \left|c_i^{(j)} - S^{(j)} + \alpha \cdot p_i\right|_{p_i} = \left|c_i^{(j)} - S^{(j)}\right|_{p_i} \quad (2)$$

Let $\left|c_i^{(j)} - S^{(j)}\right|_{p_i} = a_i^{(j)}$, then

$$\left|p_0 \cdot rand_j\right|_{p_i} = a_i^{(j)}, \quad (3)$$

for all $i \in [1, \dots, n]$. We can compute $p_0 \cdot rand_j = A^{(j)} \xrightarrow{RRNS} \left(a_1^{(j)}, a_2^{(j)}, \dots, a_n^{(j)}\right)$ with Chinese remainder theorem.

The Unique-prime-factorization theorem states that every integer greater than 1 is either prime itself or is the product of prime numbers, and this product is unique, up to the order of the factors. Hence, $p_0 \cdot rand_j = m_1^{l_1} \cdot m_2^{l_2} \cdot \dots \cdot m_t^{l_t} \leq p_0^k$, where $m_1, m_2, \dots, m_t$ are prime numbers and $l_1, l_2, \dots, l_t$ are the degree of primes, respectively.

Let $p_0^{(j)}$ is the approximation of p_0 obtained with the j data set. We calculate it as $p_0^{(j)} = \gcd\left(p_0^{(j-1)}, A^{(j)}\right)$, where $p_0^{(1)} = A^{(1)}$, $j \in [2, \dots, N]$, and N is number of pairs $(S^{(j)}, C^{(j)})$ used for p_0 approximation. From the property of the function $\gcd\left(p_0^{(j-1)}, A^{(j)}\right)$, it follows that the values of $p_0^{(j)}$ satisfy the following condition: $p_0^{(1)} \geq p_0^{(2)} \geq \dots \geq p_0^{(N)} = p_0$.

In the worst case, $p_0^{(1)} = m_1^{l_1} \cdot m_2^{l_2} \cdot \dots \cdot m_t^{l_t}$, $p_0^{(2)} = m_1^{l_1 - 1} \cdot m_2^{l_2} \cdot \dots \cdot m_t^{l_t}, \dots, p_0^{(l_1)} = m_1^{l_1 - l_1} \cdot m_2^{l_2} \cdot \dots \cdot m_t^{l_t}$, $p_0^{(l_1+1)} = m_2^{l_2 - 1} \cdot \dots \cdot m_t^{l_t}, \dots, p_0^{(l_1+l_2)} = m_2^{l_2 - l_2} \cdot \dots \cdot m_t^{l_t}$, $\dots$, $p_0^{(l_1+l_2+\dots+l_t-1)} = m_t$. Since $l_1 + l_2 + \dots + l_t - 1 \leq \lceil \mathrm{k} \cdot \log_2 p_0 \rceil$, $N \leq \lceil \mathrm{k} \cdot \log_2 p_0 \rceil$.□

The main conclusion is that the approach with secret key is more secure than with hidden moduli set of HORNS.

6. Storage Model

From the previous section, we conclude that hidden modules approach should not be used as a solution to cloud collusion problem. It leads to great data redundancy and does not ensure the security of the data. To ensure the security of data and prevent cloud collusion, we propose AC-RRNS. It is a modification of Asmooth-Bloom scheme, which is computationally secure, and decreases data redundancy by k times.

In AC-RRNS scheme, one RRNS share is stored in one cloud provider. To prevent cloud collusion, we use a secret key p_0. To reduce the data redundancy, we relax the condition $\mathrm{p}_0 < \mathrm{p}_1$ of Asmooth-Bloom using $\mathrm{p}_0 < P$.

We use RNS moduli of the size in bits equals to the size of the machine word (l bits). It is known that modules that equal to the power of two are not secure. They allow an adversary to know the piece of a confidential data equals to his data projection.

In RRNS, the computational security of the system depends on the parameters k and n. Their appropriate selection provides the necessary level of computational security. On the other hand, the properties of RNS error correction codes allow detecting and correcting errors that occur due to technical failures in data transmission and storage, or due to intentional data spoofing in collusion.

Chervyakov et al., 2017 [33] 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, however, with growing redundancy. Taking into account the volume of the data, the redundancy becomes the key factor. After the analysis of redundancy of such systems as RACS (Abu-Libdeh et al., 2010) [24], DepSky (Bessani et al., 2013) [2], etc., it is clear that the most suitable choice is $r < k$.

In our scenario, each cloud provider receives a chunk of data (share) that consists of share identifier, share properties, a projection of the original data, simplified digital signature, and moduli RNS. To generate the unique secret key, we use hash function based on SHA-3 algorithm (Pritzker & Gallagher, 2004) [39].

In proposed secret sharing scheme, we use the concept of an asymptotically perfect Asmuth-Bloom scheme with zero knowledge (Quisquater et al., 2002) [30].

Following (Gomathisankaran et al., 2011) [28], let the parameter p_0 be a secret key. The dynamic range of a system based on Asmuth-Bloom scheme is $[0, \mathrm{p}_0)$, which is not suitable to construct the method of data security in cloud computing. Moreover, the size of each share is greater than the original secret itself. It leads to more than n times increase of data volume.

To secure from cloud collusion, we combine the approaches of the two schemes: the Asmuth-Bloom and Mignotte. To formalize the proposed scheme, we use the following notation. S is a secret, $\mathrm{p}_1, \mathrm{p}_2, \dots, \mathrm{p}_\mathrm{n}$ are pairwise coprime numbers (RNS moduli set), p_0 is an integer (adaptive parameter, secret key) that is coprime with each of $\mathrm{p}_1, \mathrm{p}_2, \dots, \mathrm{p}_\mathrm{n}$.

These parameters satisfy the following three conditions.

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

where l is the length of each modulus in bits.

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

$$\forall i \in [1, \dots, n]: c_i = |S + p_0 \cdot \mathrm{rand}|_{p_i}$$

Condition 2 is the modification of Asmooth-Bloom condition $p_0 < p_1$. The dynamic range of the system is increased by $2^{\left\lfloor \log_2 \prod_{i=0}^{k-3} p_{n-i} \right\rfloor}$ times, where $2^{\left\lfloor \log_2 \prod_{i=0}^{k-3} p_{n-i} \right\rfloor} < \prod_{i=0}^{k-3} p_{n-i} \le \frac{\prod_{i=0}^{k-2} p_{n-i}}{p_1}$. Increasing dynamic range allows to process greater secret with the same size of shares. It reduces redundancy. On the other hand, Condition 2 has the property of threshold secret sharing scheme. Replacing the condition $p_0 < p_1$ in the Asmut-Bloom scheme by Condition 2 allows us to ensure the computational security and reduce the redundancy by a factor of k.

6.1 Computational security of the proposed scheme

In this section, we study the data security under the collusion. Condition 3 states that RRNS moduli set is a compact sequence. Hence, each cloud provider has approximately the same amount of information about the original data. Now, we show that AC-RRNS minimizes the probability of access to data by collusion of adversaries. To this end, we prove the following statements, corollary, and theorem.

Statement 1. *In proposed (k, n) secret sharing scheme, if an adversary coalition knows less than k secret shares and secret key p_0, then the probability obtaining the secret is less than $1/2^{(l-1)}$.*

Proof. For the set $I \subset \{1,2,\dots,n\}$ with the cardinality less than k, we can compute the value S^* that satisfies the equality $S^* = |S|_{P_I}$, where $P_I = \prod_{i \in I} p_i$. Therefore, S can be represented as: $S = S^* + P_I \cdot w$, where integer $w \in [0, \lfloor \beta / P_I \rfloor]$. Each value of w corresponds the value of C_w^* calculated by the following formula: $C_w^* = |S^* + P_I \cdot w|_{p_0}$.

Taking into account Condition 3, P_I satisfies the condition $P_I \le \prod_{i=0}^{k-2} p_{n-i}$. Consequently, the probability to compute S with the known S^*, satisfies the equality $\Pr(p(I)) \le \frac{1}{\left\lfloor \frac{\beta}{P_I} \right\rfloor} \le \frac{1}{p_{n-k+1}} < \frac{1}{2^{l-1}}$□

Statement 2. *In the proposed (k, n) scheme, if $l > k$, the probability to obtain the secret based on known k or more secret shares without secret key is less than $1/(2^{l\cdot(k-1)}(2^{l-k}-1))$.*

Proof. From Condition 2, conditions $\beta = \prod_{i=1}^{k} p_i > p_1^k$ and $\alpha = \prod_{i=0}^{k-2} p_{n-i} < (2p_1)^{k-1}$, it follows that the cardinality of the set of all possible secret keys p_0 satisfies the condition

$$\prod_{i=1}^{k} p_i - \prod_{i=0}^{k-2} p_{n-i} = \beta - \alpha > p_1^k - (2p_1)^{k-1} = p_1^{k-1}(p_1 - 2^{k-1}) > 2^{(l-1)(k-1)}(2^{l-1} - 2^{k-1}) = 2^{l(k-1)}(2^{l-k} - 1).$$

Hence, the probability to obtain p_0 is less than $1/(2^{l\cdot(k-1)}(2^{l-k}-1))$. □

Corollary 1. *If $l > k$, then the proposed (k, n) scheme satisfies Condition 2.*

Proof. Condition 2 is satisfied, if $\beta > \alpha$ or if $\beta - \alpha > 0$, hence, $\beta - \alpha > 2^{l(k-1)}(2^{l-k} - 1)$,
Since $l > k$, $2^{l-k} > 1$, following that $2^{l-k} - 1 > 0$, hence $\beta - \alpha > 0$. □

Corollary 2. *In the proposed (k, n) scheme with $l > 2k$, if the adversary coalition knows less than k secret shares and does not know secret key, then secret can be obtained with probability less then $1/2^{l\cdot k}$, which is equivalent to the brute-force method.*

Proof. From Statements 1 and 2, the probability to compute the secret without p_0 and k secret shares, following the properties of joint probability (since the events are independent), is less than

$$\frac{1}{2^{l-1}} \cdot \frac{1}{2^{l\cdot(k-1)}(2^{l-k}-1)} = \frac{1}{2^{l\cdot k-1}(2^{l-k}-1)}.$$

From Condition 1, the cardinality of the set of all possible values of the secret S satisfies the condition

$$S < p_0 < \beta < \prod_{i=1}^{k} p_i < 2^{l \cdot k}.$$

It means that probability to obtain the secret by brute-force is less then $1/2^{l \cdot k}$.

Since, $2^{l \cdot k} = 2^{l \cdot k - 1} \cdot 2 < 2^{l \cdot k - 1} \cdot (2^k - 1) < 2^{l \cdot k - 1} \cdot (2^{l-k} - 1)$, this probability is higher than the probability to find the secret with the incomplete set of shares.

Therefore, the probability to obtain the secret is equal to

$$\max\left\{\frac{1}{2^{l \cdot k}}, \frac{1}{2^{l \cdot k - 1} \cdot (2^{l-k} - 1)}\right\} = \frac{1}{2^{l \cdot k}}$$

Hence, in the proposed (k, n) scheme with $l > 2k$, the complexity of obtaining the secret for an adversary coalition with known less than k secret shares and unknown secret key is equivalent to the complexity of brute-force method. □

Now, we show the computational security of the proposed scheme. The concept of computational security is based on the following idea: Information cannot be effectively restored if there is no complete information. Therefore, the scheme is computationally secure, if the adversary knows the secrets $S^{(1)}$, $S^{(2)}$ and incomplete sets of shares $C^{(1)}$, $C^{(2)}$, but cannot map $\left(S^{(1)}, C^{(1)}\right)$ and $\left(S^{(2)}, C^{(2)}\right)$, unambiguously.

Computational security for secret sharing schemes can be defined in more strong way (Krawczyk, 1993) [25]. It is based on the polynomial indistinguishability concept (Goldreich, 1993) [53]. For any probability distribution $D(C, S)$, a secret sharing scheme is computationally secure if, for any pair of secrets $S^{(1)}$, $S^{(2)}$ and incomplete subsets of shares $C^{(1)}$ and $C^{(2)}$, the distributions $D\left(C^{(1)}, S^{(1)}\right)$ and $D\left(C^{(1)}, S^{(1)}\right)$ are polynomial indistinguishable, i.e. for any probabilistic algorithm A

$$\left|\Pr\left(A\left(D\left(C^{(1)}, S^{(1)}\right)\right) = 1\right) - \Pr\left(A\left(D\left(C^{(2)}, S^{(2)}\right)\right) = 1\right)\right| < \frac{1}{poly(n, k)},$$

where $poly(n, k)$ is the some polynomial over the amount of possible shares.

Theorem 2. *The proposed scheme is computationally secure.*

Proof. To prove the computational security of proposed scheme, we use the auxiliary inequality.

$$\forall\, a, b, c \in R\colon |a - b| \le |a - c| + |b - c| \tag{4}$$

Let $a = \Pr\left(A\left(D\left(C^{(1)}, S^{(1)}\right)\right) = 1\right)$, $b = \Pr\left(A\left(D\left(C^{(2)}, S^{(2)}\right)\right) = 1\right)$, $c = \Pr\left(D\left(C^{(1)}, S^{(1)}\right) = 1\right)$. We get:

$$\begin{aligned}\left|\Pr\left(\mathrm{A}\left(\mathrm{D}\left(\mathrm{C}^{(1)}, \mathrm{S}^{(1)}\right)\right) = 1\right) - \Pr\left(\mathrm{A}\left(\mathrm{D}\left(\mathrm{C}^{(2)}, \mathrm{S}^{(2)}\right)\right) = 1\right)\right| \\ \le \left|\Pr\left(\mathrm{A}\left(\mathrm{D}\left(\mathrm{C}^{(1)}, \mathrm{S}^{(1)}\right)\right) = 1\right) - \Pr\left(\mathrm{D}\left(\mathrm{C}^{(1)}, \mathrm{S}^{(1)}\right) = 1\right)\right| \\ + \left|\Pr\left(\mathrm{A}\left(\mathrm{D}\left(\mathrm{C}^{(2)}, \mathrm{S}^{(2)}\right)\right) = 1\right) - \Pr\left(\mathrm{D}\left(\mathrm{C}^{(1)}, \mathrm{S}^{(1)}\right) = 1\right)\right|\end{aligned} \tag{5}$$

where $\Pr\left(D\left(C^{(1)}, S^{(1)}\right) = 1\right)$ is the to probability of obtaining the secret using the first k shares.

Since the number of desired outcomes is less than or equal to $\prod_{i=1}^{k} p_i$ and the total number of all outcomes is $\prod_{i=1}^{n} p_i$, then the probability is

$$\Pr\left(\mathrm{A}\left(\mathrm{D}\left(\mathrm{C}^{(1)}, \mathrm{S}^{(1)}\right)\right) = 1\right) \le \frac{\prod_{i=1}^{k} p_i}{\prod_{i=1}^{n} p_i} = \frac{1}{\prod_{i=k+1}^{n} p_i},$$

$$\Pr\left(\mathrm{A}\left(\mathrm{D}\left(\mathrm{C}^{(2)}, \mathrm{S}^{(2)}\right)\right) = 1\right) \le \frac{\prod_{i=1}^{k} p_i}{\prod_{i=1}^{n} p_i} = \frac{1}{\prod_{i=k+1}^{n} p_i},$$

$$\Pr\left(\mathrm{D}\left(\mathrm{C}^{(1)}, \mathrm{S}^{(1)}\right) = 1\right) = \frac{1}{\prod_{i=1}^{k} p_i}.$$

From Condition 3, it follows that $p_0^k < \prod_{i=1}^{k} p_i < 2^k p_0^k$ and $p_0^{n-k} < \prod_{i=k+1}^{n} p_i < 2^{n-k} p_0^{n-k}$.

Therefore, $\frac{1}{2^k p_0^k} < \frac{1}{\prod_{i=1}^{k} p_i} < \frac{1}{p_0^k}$ and $\frac{1}{2^{n-k} p_0^{n-k}} < \frac{1}{\prod_{i=k+1}^{n} p_i} < \frac{1}{p_0^{n-k}}$

Let us estimate terms of (5)

$$\left|\Pr\left(\mathrm{A}\left(\mathrm{D}\left(\mathrm{C}^{(1)},\mathrm{S}^{(1)}\right)\right)=1\right)-\Pr\left(\mathrm{D}\left(\mathrm{C}^{(1)},\mathrm{S}^{(1)}\right)=1\right)\right|<\max\left\{\frac{1}{\mathrm{p}_0^{\mathrm{n-k}}}-\frac{1}{2^{\mathrm{k}}\mathrm{p}_0^{\mathrm{k}}},\frac{1}{\mathrm{p}_0^{\mathrm{k}}}-\frac{1}{2^{\mathrm{n-k}}\mathrm{p}_0^{\mathrm{n-k}}}\right\}$$
$$\left|\Pr\left(\mathrm{A}\left(\mathrm{D}\left(\mathrm{C}^{(2)},\mathrm{S}^{(2)}\right)\right)=1\right)-\Pr\left(\mathrm{D}\left(\mathrm{C}^{(1)},\mathrm{S}^{(1)}\right)=1\right)\right|<\max\left\{\frac{1}{\mathrm{p}_0^{\mathrm{n-k}}}-\frac{1}{2^{\mathrm{k}}\mathrm{p}_0^{\mathrm{k}}},\frac{1}{\mathrm{p}_0^{\mathrm{k}}}-\frac{1}{2^{\mathrm{n-k}}\mathrm{p}_0^{\mathrm{n-k}}}\right\}. \quad (6)$$

By substituting (6) in (5), we get:

$$\left|\Pr\left(A\left(D\left(C^{(1)},S^{(1)}\right)\right)=1\right)-\Pr\left(A\left(D\left(C^{(2)},S^{(2)}\right)\right)=1\right)\right|<2\cdot\max\left\{\frac{1}{p_0^{n-k}}-\frac{1}{2^k p_0^k},\frac{1}{p_0^k}-\frac{1}{2^{n-k}p_0^{n-k}}\right\}. \quad (7)$$

It means that the proposed scheme satisfies the formal definition of computational security.□

Theorem 2 has a significant practical importance. In particular, it states that the adversary does not have any additional information from an incomplete set of shares.

Let $(S^{(1)},C^{(1)})$ and $(S^{(2)},C^{(2)})$ satisfy the following assertions for all $i\in[1,\dots,n]$:

$$c_i^{(1)}=\left|S^{(1)}+p_0\cdot rand_1\right|_{p_i},\ c_i^{(2)}=\left|S^{(2)}+p_0\cdot rand_2\right|_{p_i}. \quad (8)$$

Since for all $i\in[1,\dots,n]$ $\gcd(p_0,p_i)=1$, there exist $rand_1'$, $rand_2'$, p_0', such that the following equations are satisfied:

$$c_i^{(1)}=\left|S^{(2)}+p_0'\cdot rand_2'\right|_{p_i}.,\ c_i^{(2)}=\left|S^{(1)}+p_0'\cdot rand_1'\right|_{p_i}. \quad (9)$$

From (8) and (9), it follows that to unambiguously map $(S^{(1)},C^{(1)})$ and $(S^{(2)},C^{(2)})$, p_0 is required. Since p_0 is not known, the our scheme is computationally secure.

Example 2. Let the parameters of the secret sharing scheme be $k=2$, $n=4$, and RRNS moduli set is $p_1=3221225473, p_2=3221225479, p_3=3221225533, p_4=3221225549$. The secret key is $p_0=2635968733367020$. Let $S^{(1)}=6323947392560$ and $S^{(2)}=51771684174750$ be two secret messages, $rand_1=15$, and $rand_2=6$.

If we use $p_0'=9089547356438$ as a secret key and $rand_1'=4345$, $rand_2'=1745$, then the values $S^{(1)}$ and $S^{(2)}$ are exchanged and, therefore, it is not possible to unambiguously map ($S^{(1)},C^{(1)}$) and $(S^{(2)},C^{(2)})$.

This example demonstrates that the scheme is computationally secure.

7. Analysis

In this section, we present a comparative analysis of the proposed scheme with Asmooth-Bloom and Mignotte schemes.

7.1. Data redundancy

In the worst case, the number of bits needed to store $(p_1-1,p_2-1,\dots,p_n-1)$ is approximately equal to $\sum_{i=1}^{n}\lceil\log_2 p_i\rceil$.

The length of input data approximately equals to $\lceil\log_2 p_0\rceil$. We calculate redundancy as the ratio of the stored encoded data and original data size:

$$\frac{\sum_{i=1}^{n}\lceil\log_2 p_i\rceil}{\lceil\log_2 p_0\rceil}=\frac{n\cdot l}{\lceil\log_2 p_0\rceil}$$

Since $\lceil\log_2 p_0\rceil$ satisfies the inequality$(k-1)(l-1)<\lceil\log_2 p_0\rceil\le k\cdot l$, the data redundancy satisfies the inequality:

$$\frac{n}{k}\le\frac{\sum_{i=1}^{n}\lceil\log_2 p_i\rceil}{\lceil\log_2 p_0\rceil}<\frac{n\cdot l}{(k-1)(l-1)}$$

We show the relation between redundancy of the scheme and the parameters with $l=32$ (Fig. 1).

From Fig. 1, we conclude that the proposed scheme has greater data redundancy than Mignotte scheme and less than semantic secure Asmoth-Bloom scheme. As it shown above, proposed AC-RRNS scheme is computationally secure while Mignotte scheme considered insecure (Quisquater et al., 2002) [30].

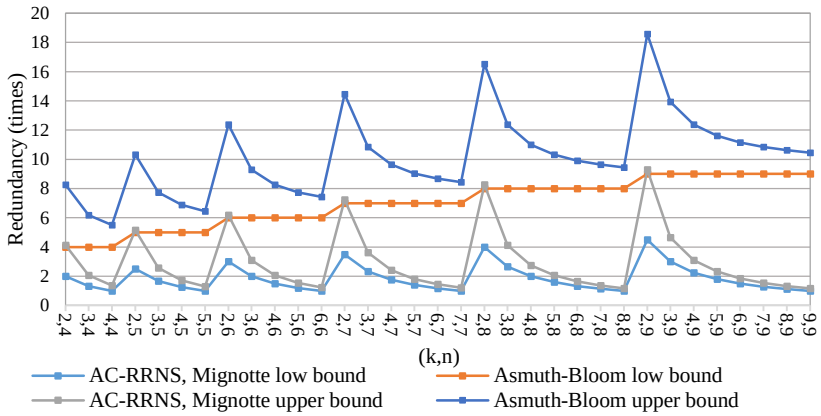


Fig. 1. Data redundancy with $l = 32$

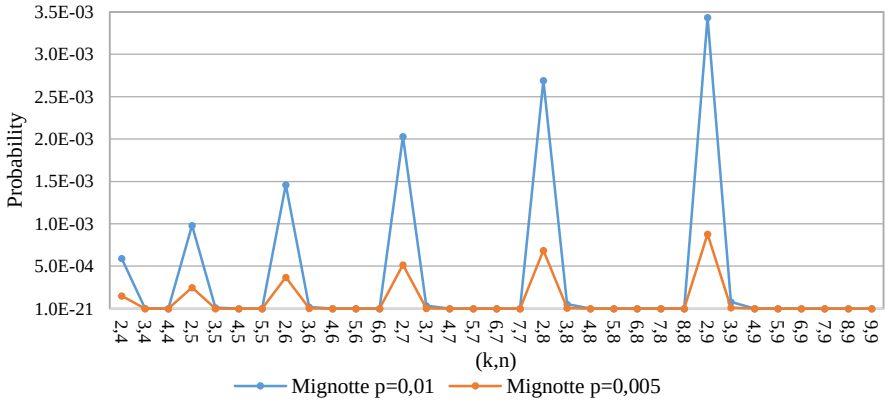
7.2. Probability of collusion

Let the probability of collusion of any two adversaries be p. The probability of coalition of k or more members can be computed with the formula:

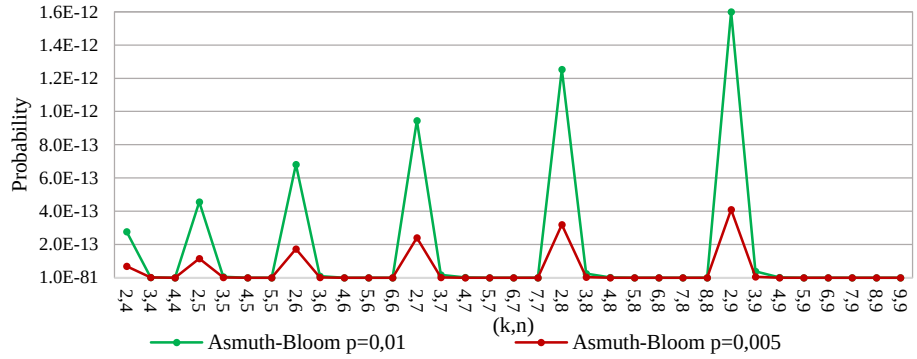
$$\sum_{i=k}^{n} C_n^i p^i (1-p)^{n-i}$$

The probability of gaining access to the data is $1/(2^{l\cdot(k-1)}(2^{l-k}-1))$ (See Statement 2**).** Hence, the probability P_C of coalition creation that can get unauthorized access can be computed using the product rule:

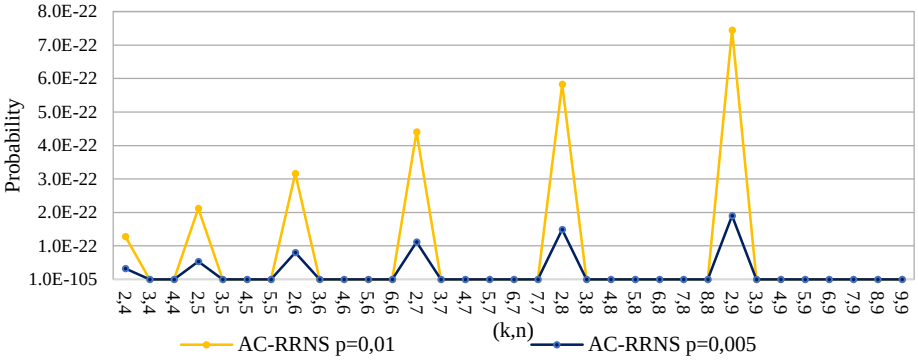
$$P_C = \frac{1}{2^{l\cdot(k-1)}(2^{l-k}-1)} \sum_{i=k}^{n} C_n^i p^i (1-p)^{n-i}.$$



(a) Mignotte scheme



(b) Asmuth-Bloom



(c) AC-RRNS

Fig. 2. Probability of unauthorized access by coalition with k adversaries versus (k,n), $l = 32$

Fig. 2 shows the probability P_C of obtaining a secret by adversary coalition with k participants. We see that in AC-RRNS the probability is significantly less than in the Mignotte and Asmuth-Bloom schemes. The highest probabilities are $3.5E^{-03}$, $1.6E^{-12}$, $8.0E^{-22}$ in Mignotte, Asmuth-Bloom, and AC-RRNS, respectively. The lowest probabilities are $1.0E^{-21}$, $1.0E^{-81}$, $1.0E^{-105}$ in Mignotte, Asmuth-Bloom, and AC-RRNS, respectively.

Table 3
Main properties of the studied schemes

	$< k$ *secret shares*	$\geq k$ *secret shares*	Secret key	Obtaining secret probability (upper bound)	Redundancy (low bound)	Dynamic range (upper bound)	Perfect	Asymptotically ideal	Computationally secure	Anti-Collusion
Asmuth-Bloom	•		•	$1/2^{l-1}$	n	p_1	•	•		•
		•		$1/2^{l-1}$						
	•			$1/2^{l-1}$						
Mignotte		•		1	$\frac{n}{k}$	$\prod_{i=1}^{k} p_i$				
	•			$1/2^{l-1}$						
AC-RRNS	•		•	$1/2^{l-1}$	$\frac{n}{k}$	$\prod_{i=1}^{k} p_i$	•		•	•
		•		$1/(2^{l\cdot(k-1)}(2^{l-k}-1))$						
	•			$1/2^{l\cdot k}$						

Table 3 shows main properties of these schemes. It is clear that AC-RRNS has better characteristics. Together with Asmuth-Bloom, AC-RNNS is able to solve cloud collusion problem. However, AC-RNNS is computationally secure. The probability of obtaining the secret in the scenario, where the adversary coalition does not know the secret key is much lower. Moreover, AC-RNNS has significantly lower redundancy that is important for real systems.

6. Conclusion

Cloud computing has become an integrated part of IT landscapes. However, new technologies also pose new problems. For instance, the uncertainty of failures and varied threats play a key role in security of distributed online cloud storage systems. We identify three main security threats: deliberate, accidental, and collusion. In this paper, we focus on collusion problem, when adversaries can get access to the parts of confidential data in one or several storages. We qualify the methods used to solve the collusion according to functional characteristics: collusion detection, collusion prevention, and data security protection in case of collusion.

We prove that known homomorphic encryption scheme HORNS for cloud computing is not computationally secure. As a result, adversary coalition enables to obtain the secret in the hidden RRNS modules framework. We prove that the schemes with a secret key are more secured than with hidden moduli set in HORNS.

We introduce new AC-RRNS configurable data storage scheme based on the computational secure and reliable secret sharing scheme based on RRNS and study its security features. We solve the problem of cloud collusion by the simultaneous use of the ideas behind the Mignotte and asymptotically ideal Asmuth-Bloom secret sharing schemes. To reduce the uncertainty of risks of data security breaches and denial of access to data, we use error localization and correction codes of RRNS.

We consider three scenarios when the adversary coalition knows and does not know the required number of secret shares, and knows and does not know the secret key. The probabilities to obtain the secret for each scenario are provided. We prove that they are much lower in AC-RNNS comparing with Mignotte and Asmuth-Bloom.

We demonstrate that the proposed scheme ensures security under several types of attacks. We propose approaches to selection of the parameters of AC-RRNS secret sharing scheme to optimize the system behavior and reduce redundancy of the encrypted data.

Further study is required to assess multi-parameter efficiency of software and hardware implementations. This will be the subject of future work.

Acknowledgements

The work is partially supported by Russian Foundation for Basic Research (RFBR) 18-07-01224-a, 18-07-00109, State task No. 2.6035.2017, and Russian Federation President Grants SP-1215.2016.5, SP-2236.2018.5, and MK-6294.2018.9.

References

1. A. Barg, G. Zemor, Distance properties of expander codes, IEEE T Inform. Theory. 52 (1) (2006) 78-90.
2. A. Bessani, M. Correia, B. Quaresma, F. André, P. Sousa, DepSky: dependable and secure storage in a cloud-of-clouds, ACM T. Storage 9 (4) (2013) 12.
3. A. Celesti, M. Fazio, M. Villari, A. Puliafito, Adding long-term availability, obfuscation, and encryption to multi-cloud storage systems, J. Netw. Comput. Appl. 59 (2016) 208-218.
4. A. Joux, K. Nguyen, Separating decision Diffie–Hellman from computational Diffie–Hellman in cryptographic groups, J. Cryptol. 16 (4) (2003) 239-247.
5. 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 Gener. Comp. Sy. 28 (7) (2012) 965-976.
6. 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: Proceedings in the 28th International Workshop on Database and Expert Systems Applications (DEXA), 2017, pp. 137-141.
7. A. Tchernykh, U. Schwiegelsohn, E. Talbi, M. Babenko, Towards Understanding Uncertainty in Cloud Computing with risks of Confidentiality, Integrity, and Availability, J. Comput. Sci. 2016. doi: 10.1016/j.jocs.2016.11.011
8. A.C. Mora, Y. Chen, A. Fuchs, A. Lane, R. Lu, P. Manadhata, Cloud Security Alliance, Top ten big data security and privacy challenges. https://downloads.cloudsecurityalliance.org/initiatives/bdwg/Big_Data_Top_Ten_v1.pdf, 2017 (accessed 23 December 2017).
9. B. K. Samanthula, G. Howser, Y. Elmehdwi, S. Madria, An efficient and secure data sharing framework using homomorphic encryption in the cloud, in: Proceedings of the 1st International Workshop on Cloud Intelligence, 2012, p. 8.
10. B. Schneier, Homomorphic Encryption Breakthrough, Schneier on Security. https://www.schneier.com/blog/archives/2009/07/homomorphic_enc.html, 2017 (accessed 23 December 2017).

11. C. H. Chang, A. S. Molahosseini, A. A. E. Zarandi, T. F. Tay, Residue number systems: A new paradigm to datapath optimization for low-power and high-performance digital signal processing applications, IEEE Circ. Syst. Mag. 15 (4) (2015) 26-44.
12. C. Ye, H. Ling, Z. Xiong, F. Zou, C. Liu, F. Xu, Secure Social Multimedia Big Data Sharing Using Scalable JFE in the TSHWT Domain, ACM T. Multim. Comput. 12 (4s) (2016) 61.
13. D. A. Fernandes, L. F. Soares, J. V. Gomes, M. M. Freire, P. R. Inácio, Security issues in cloud environments: a survey, Int. J. Inf. Secur. 13 (2) (2014) 113-170.
14. D. Boneh, J. Shaw, Collusion-secure fingerprinting for digital data, IEEE T. Inform. Theory. 44 (5) (1998) 1897-1905.
15. D. Hubbard, M. Sutton, Cloud Security Alliance, Top threats to cloud computing v1. 0. https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf, 2017 (accessed 23 December 2017).
16. D. Kliazovich, J. E. Pecero, A. Tchernykh, P. Bouvry, S. U. Khan, and A. Y. Zomaya, CA-DAG: Modeling Communication-Aware Applications for Scheduling in Cloud Computing, J. Grid Comput. 14(1) (2016) 23-39.
17. D. S. Phatak, S. D. Houston, New distributed algorithms for fast sign detection in residue number systems (RNS), J. Parallel Distr. Com. 97 (2016) 78-95.
18. G. Ateniese, K. Fu, M. Green, S. Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage, ACM T. Inform. Syst. Se. 9 (1) (2006) 1–30.
19. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, K. Ramchandran, Network Coding for Distributed Storage Systems, EEE T. Inform. Theory. 56 (9) (2010) 4539-4551.
20. G. Dimauro, S. Impedovo, G. Pirlo, A new technique for fast number comparison in the residue number system, IEEE T. Comput. 42 (5) (1993) 608-612.
21. G. Dimauro, S. Impedovo, R. Modugno, G. Pirlo, R. Stefanelli, Residue-to-binary conversion by the" quotient function, IEEE T. Circuits-II 50(8) (2003) 488-493.
22. G. Pirlo, D. Impedovo, A new class of monotone functions of the residue number system, Int. J. Math. Models Methods Appl. Sci, 7(9) (2013) 803-809.
23. G. Zémor, On expander codes, IEEE T. Inform. Theory 47 (2) (2001) 835-837.
24. 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.
25. H. Krawczyk, Secret Sharing Made Short, in: Proceedings in the 13th Annual International Cryptology Conference, vol. 93, 1993, pp. 136-146.
26. J. Hur, Improving security and efficiency in attribute-based data sharing, IEEE T Knowl. Data En. 25 (10) (2013) 2271-2282.
27. Kaspersky DDoS Intelligence Report for Q1 2016. https://securelist.com/kaspersky-ddos-intelligence-report-for-q1-2016/74550/, 2017 (accessed 23 December 2017).
28. M. Gomathisankaran, A. Tyagi, K. Namuduri, HORNS: A homomorphic encryption scheme for Cloud Computing using Residue Number System, in: Proceedings in the 45th Annual Conference on Information Sciences and Systems (CISS), 2011, pp. 1-5.
29. M. Jensen, J. Schwenk, N. Gruschka, L. L. Iacono, On technical security issues in cloud computing, in: Proceedings in the IEEE International Conference on Cloud Computing, 2009, pp. 109–116.
30. M. Quisquater, B. Preneel, J. Vandewalle, On the security of the threshold scheme based on the Chinese remainder theorem, in: Proceedings in the International Workshop on Public Key Cryptography, 2002, pp. 199-210.
31. M. Sipser, D. A. Spielman, Expander codes, IEEE T. Inform. Theory. 42 (6) (1996) 1710-1722.
32. N. Burgess, Scaling an RNS number using the core function, in: Proceedings of the 16th IEEE Symposium on Computer Arithmetic, 2003, pp. 262-269.
33. 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, Future Gener. Comp. Sy. 2017. doi:10.1016/j.future.2017.09.061
34. N. I. Chervyakov, A. S. Molahosseini, P. A. Lyakhov, M. G. Babenko, , M. A. Deryabin, Residue-to-binary conversion for general moduli sets based on approximate Chinese remainder theorem, Int. J. Comput. Math. 94 (9) (2017) 1833-1849.
35. N. S. Szabo, R. I. Tanaka, Residue arithmetic and its applications to computer technology, McGraw-Hill, 1967.
36. OpenFog Reference Architecture for Fog Computing. https://www.openfogconsortium.org/wp-content/uploads/OpenFog_Reference_Architecture_2_09_17-FINAL.pdf, 2017 (accessed 23 December 2017).
37. P. A. Mohan, RNS to binary conversion using diagonal function and Pirlo and impedovo monotonic function, Circ. Syst. Signal. Pr. 35 (3) (2016) 1063-1076.
38. P. Patronik, S. J. Piestrak, Design of Reverse Converters for General RNS Moduli Sets and (even), IEEE T. Circuits-I 61 (6) (2014) 1687-1700.

39. P. Pritzker, P. Gallagher, Information Tech Laboratory National Institute of Standards and Technology, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. https://www.nist.gov/publications/sha-3-standard-permutation-based-hash-and-extendable-output-functions, 2017 (accessed 23 December 2017).
40. S. Bi, W. J. Gross, The mixed-radix Chinese remainder theorem and its applications to residue comparison, IEEE T. Comput. 57 (12) (2008) 1624-1632.
41. S. Chessa, R. Di Pietro, P. Maestrini, Dependable and secure data storage in wireless ad hoc networks: an assessment of DS^2, in: Proceedings in the First IFIP TC6Working Conference, 2004, pp. 184-198.
42. S. G. R. Ekodeck, R. Ndoundam, PDF steganography based on Chinese Remainder Theorem, J. Inf. Sec. and Appl. 29 (2016) 1-15.
43. S. Kianpisheh, S. Jalili, N. Charkari, Predicting Job Wait Time in Grid Environment by Applying Machine Learning Methods on Historical Information, Int. J. Grid Distr. Com. 5 (3) (2012)11-22.
44. S. Subashini, V. Kavitha, A survey on security issues in service delivery models of cloud computing, J. Netw. Comput. Appl. 34 (1) (2011) 1-11.
45. T. Ristenpart, E. Tromer, H. Shacham, S. Savage, Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds, in: Proceedings of the 16th ACM conference on Computer and communications security, 2009, pp. 199-212.
46. T. Van Vu, Efficient implementations of the Chinese remainder theorem for sign detection and residue decoding, IEEE T. Comput. 100 (7) (1985) 646-651.
47. Y. Fu, B. Sun, A scheme of data confidentiality and fault-tolerance in cloud storage, in: Proceedings of the IEEE 2nd International Conference on Cloud Computing and Intelligent Systems (CCIS), vol. 1, 2012, pp. 228-233.
48. Y. Li, V. Swarup, S. Jajodia, Fingerprinting relational databases: Schemes and specialties, IEEE T. Depend. Secure. 2 (1) (2005) 34-45.
49. Y. Wang, Residue-to-binary converters based on new Chinese remainder theorems, IEEE T. Circuits-Ii 47 (3) (2000) 197-205.
50. Yu. N. Sotskov, F. Werner, Sequencing and Scheduling with Inaccurate Data, Nova Science Pub, Applied Statistica Science, Minsk, 2014.
51. Z. Xiao, Y. Xiao, Security and privacy in cloud computing, IEEE Commun. Surv. Tut. 15(2) (2013) 843-859.
52. Z. Zhu, R. Jiang, A secure anti-collusion data sharing scheme for dynamic groups in the cloud, IEEE T Parall. Distr. 27 (1) (2016) 40-501.
53. O. Goldreich, A uniform-complexity treatment of encryption and zero-knowledge, J. Cryptol. 6 (1), (1993) 21-53.
54. S. Kirthica, R. Sridhar, A residue-based approach for resource provisioning by horizontal scaling across heterogeneous clouds, Int. J. Approx. Reason. (2018)
55. A. Omri, K. Benouaret, D. Benslimane, M. N. Omri, Towards an understanding of cloud services under uncertainty: A possibilistic approach, Int. J. Approx. Reason. 98, (2018) 146-162.
56. A. Tchernykh, V. Miranda-López, M. Babenko, F. Armenta-Cano, G. Radchenko, A. Yu. Drozdov, A, Avetisyan, Performance Evaluation of Secret Sharing Schemes with Data Recovery in Secured and Reliable Heterogeneous Multi-Cloud Storage, Cluster Comput. 2018.
57. A. Tchernykh, D. Trystram, C. Brizuela, I. Scherson, Idle Regulation in Non-Clairvoyant Scheduling of Parallel Jobs, Discrete Appl. Math. 157 (2009) 364–376.
58. V. Miranda-López, J. M. Cortés-Mendoza, A. Tchernykh, M. Babenko, G. Radchenko, S. Nesmachnow, Z. Du, Experimental Analysis of Secure and Reliable Schemes for Cloud Storage based on RNS, in: High Performance Computing. CARLA 2017 (CCIS), vol. 796, 2018, pp. 370-383.
59. A. Tchernykh, J. M. Cortés-Mendoza, A. Feoktistov, I. Bychkov, L. Didelot, P. Bouvry, G. Radchenko, K. Borodulin, Configurable Cost-Quality Optimization of Cloud-based VoIP, J. Parallel Distr. Com. (2018)
60. R. Massobrio, S. Nesmachnow, A. Tchernykh, A. Avetisyan, G. Radchenko, Towards a Cloud Computing Paradigm for Big Data Analysis in Smart Cities, Program. Comput. Soft. 44(3), (2018) 181–189