Secure Verifiable Secret Short Sharing Scheme for Multi-Cloud Storage

Maxim Deryabin[1],
Nikolay Chervyakov[2]
North-Caucasus Federal University
Stavropol, Russia
{[1]maderiabin,[2]ncherviakov}@ncfu.ru

Andrei Tchernykh[3,*]
CICESE Research Center, Ensenada, Mexico
South Ural State University, Chelyabinsk, Russia
Institute for System Programming of the RAS
Moscow, Russia, [3]chernykh@cicese.mx

Mikhail Babenko[4],
Nikolay Kucherov[5]
North-Caucasus Federal University
Stavropol, Russia
{[4]mgbabenko,[5]nkucherov}@ncfu.ru

Vanessa Miranda-López[6]
CICESE Research Center
Ensenada, BC, México
[6]vmiranda@cicese.edu.mx

Arutyun Avetisyan[7]
Institute for System Programming of the Russian
Academy of Sciences, Moscow, Russia
[7]arut@ispras.ru

Abstract—**In this paper, we propose a new approach to the construction of computationally secure secret sharing scheme based on the simultaneous use of the Residue Number System (RNS), as the symmetric encryption, and the perfect Asmuth-Bloom secret sharing scheme. We combine the useful features of Redundant RNS to design space efficient secret sharing scheme with enough level of security and to control of data integrity. This combination provides verifiability and achieves a high speed of data processing.**

Keywords – Threshold secret sharing scheme, computationally secure secret sharing, verifiable scheme, Residue Number System, cloud data storage, distributed storage, multi-clouds.

I. Introduction

Distributed cloud storage systems are actively used in the design of IT solutions. The volumes of stored data and the number of users are constantly increasing. However, according to security report of Cloud Security Alliance 2016 [4], there is a high probability of a breach of confidentiality and integrity.

An important part of the Quality of Service (QoS) of cloud and distributed storage is the availability of data. Errors can occur during writing, reading, storage, transmission, and processing. They introduce changes to the original data, possible loss or distortion. Moreover, storage service providers can access the data. Data can be corrupted or revealed by hacker attacks.

The violation of availability can be caused by malfunctions of equipment and network, as well as by other factors related, for example, to the termination of the service or bankruptcy of the provider.

To solve the data confidentiality, availability and integrity issues many approaches are used [1-3]. The main drawback of the most of them is the lack of a unified approach. Very often, a single approach solves one or two issues but requires a combination of other approaches. This leads to a complication of the system, which may be unacceptable in practice. For example, symmetric and asymmetric encryption allows to ensure confidentiality but does not affect the integrity and availability of information. On the other hand, erasure codes [5] increase the reliability of data storage, but it is not enough to protect data from unauthorized access.

An important trend is to use multi-cloud storage systems [6-8]. In such systems, data is distributed among different cloud providers. Due to the partially redundant storage of data using the resources of different cloud storage services, it is possible to achieve a high level of data availability and integrity, since the probability of simultaneous failure of several cloud services at once is quite small [3]. A similar idea is used for distributed file systems, such as Hadoop [9], GFS [10], etc.

The idea of multi-cloud storage is widely used in scientific researches. DepSky system [1] uses four different cloud providers with ensuring confidentiality, availability, and integrity of data. Another important example is the method of distribution of large medical data using multi-cloud approach [11].

The basis of the algorithms is the Secret Sharing Schemes (SSS) [12-13]. The SSS is a method for representing data in a distributed form as a set of shares. The reconstruction of data in its original form is possible only if certain conditions are met.

In Threshold Secret Sharing Schemes (TSSS), the condition is the availability of a certain number of shares together. This number must exceed a predetermined threshold. Each share is associated with a single cloud provider or a distributed system node. It is assumed that this provider or node does not have access to the other shares. This allows ensuring confidentiality because each share does not carry enough information about the initial data.

[*] Corresponding author

978-1-5386-7879-4/18/$31.00 ©2018 IEEE
DOI 10.1109/HPCS.2018.00114

Considering redundancy and processing speed, the most advantageous SSS is Computationally Secure SSS (CS SSS). Such a scheme proposed by Krawczyk [14] has widespread use in the cloud data storages [2, 8, 11, 15].

In this paper, we propose a new approach to the construction of computationally secure secret sharing scheme. It is based on the simultaneous use of the Residue Number System (RNS) [16-17], as an approach for dispersal of data, symmetric encryption, and the perfect Asmuth-Bloom SSS to ensure the security of encryption key distribution [18]. Besides, the proposed solution provides the possibility to verify the correctness of stored data by using the correcting properties of the RNS without using additional algorithms and without significant increase of the overhead for data representation.

RNS is a non-positional numeral system that allows representing the data in distributed form by operations on residual classes. Both the Perfect Secret Sharing Schemes (PSSS) [18] and Information Dispersal Algorithms (IDA) [19] are constructed based on RNS. Furthermore, efficient error correcting codes [17, 20-21] are proposed based on the RNS, which are important components of data integrity control. Our proposal is based on the structural simplicity of the RNS code, which makes data encoding and decoding operations efficient.

The paper is structured as follows. The next section briefly reviews related works, models, and algorithms for confidentiality, integrity and availability problems. Section 3 describes the main concepts of redundant residue number system. Section 4 discusses properties of secret sharing schemes. Section 5 presents our computationally secure verifiable SSS based on RNS. Section 6 provides an experimental analysis. Section 7 concludes with a summary of our study.

II. Related work

Storing data in public cloud storage requires special attention to confidentiality, integrity, and availability. It is possible to single out several methods used in practice, which provide all three components of the reliable storage partially or completely. It is necessary to take into account the peculiarities of the architecture of many-cloud storage systems. An important metric is the redundancy of the chosen approach, which affects the cost and complexity of the system being designed.

Table I shows the main types of approaches that can be used to organize multi-cloud storage. Data replication is used in practice [9-10] and provides availability and data integrity by its duplication on each node of the system. This approach leads to a very high level of redundancy, which may be unacceptable for a multi-cloud storage model because of the higher cost of resources in comparison with local distributed storage. Encryption schemes solve only part of the problem associated with data confidentiality.

However, data replication [10], erasure codes [22], error correction codes [23] and Information dispersal algorithms [24] cannot be considered as mechanisms for ensuring confidentiality. IDA and erasure codes ensure a high level of data availability and, under some assumptions [25], data integrity. Block error correction codes, such as Reed-Solomon codes [26], ensure data integrity, but require large overhead for organizing data storage and processing.

TABLE I. CHARACTERISTICS OF METHODS OF PROVIDING RELIABILITY OF DATA STORAGE IN CLOUD SYSTEMS

Method	**Confidentiality**	**Integrity**	**Availability**	**Redundancy**
Replication		●	●	very high
Encryption Schemes	high			low
Erasure Codes			●	medium
Error Correction Codes		●	●	high
Information Dispersal Algorithms	low		●	medium
Perfect Secret Sharing Schemes	high		●	very high
Computationally Secure SSS	medium		●	medium

SSSs combine the properties of information dispersal schemes and cryptographic systems. Therefore, they have the best capabilities for ensuring confidentiality, integrity, and availability of data. This is confirmed by using SSS for the organization of multi-cloud data storage [2-3, 7-8, 11, 13].

Perfect SSS [27-28] provides a high level of data confidentiality but require high redundancy. Verifiable SSSs, which also allow controlling data integrity [29], have high redundancy. Among the perfect schemes for separating the secret are the Shamir scheme [12] based on polynomial interpolation, the Asmut-Bloom scheme[18, 30] based on the RNS, and the Blackley scheme [31] based on intersections of multidimensional hyperplanes of some geometric space.

The PSSSs are considered as a perfect secure in the cryptographic sense, where the security means the inability to recover the initial secret with an insufficient number of shares.

All these schemes have one common drawback: they lead to significant redundancy. Most often, SSSs are used for the distribution of secret keys. However, high redundancy makes them inapplicable in the multi-cloud storage.

One of the ways to reduce redundancy is computationally secure secret sharing schemes [14]. The most common scheme is proposed by Krawczyk [14]. It is a combination of symmetric encryption, data distribution by the Rabin's IDA [24], and perfect secret sharing by Shamir [12] to store the encryption key along with the data. This approach has almost the same redundancy as the Rabin scheme. Computationally secure threshold SSSs guarantee that the secret cannot be recovered in an acceptable time if the adversary has an insufficient amount of data [14].

The Rabin scheme achieves the minimum possible redundancy to ensure the availability of data and, at the same time, provide an acceptable level of confidentiality. Integrity analysis can be implemented, for example, by replacing the Rabin scheme with a structurally similar Reed-Solomon code [26]. Such a scheme can ensure a maximum possible level of data verification and integrity.

However, the main disadvantage of the Rabin scheme is the execution of matrix calculations in the Galois field $GF(2^w)$, where w should be large enough to provide the necessary cryptographic security.

The CS-SSS proposed in this paper is devoid of this drawback. Instead of the Rabin scheme, we propose to use RNS. It avoids many complex operations by replacing matrix computations over the polynomial finite field by modular reduction.

Redundant Residue Number System (RRNS) is a convenient tool for designing SSSs and error correction codes at the same time [17]. Recently, RRNS is applied as a secret sharing scheme for various applications, including cloud data storage [3, 7, 32-34].

In this paper, we propose a verifiable computationally stable SSS based on the encryption of data and their representation in residual classes. Using RNS instead of the Rabin scheme leads to the same redundancy, but significantly increases the speed of encoding and decoding of data. Also, the correction properties of RNS are used to implement data integrity control.

III. Redundant Residue Number System

RNS is a non-positional numeral system, where every number X represented as a tuple of k residues x_i by division X modulo m_i from moduli set:

$$X = (x_1, x_2, \dots, x_k),\ x_i = X \bmod m_i,\ i = 1,2,\dots,k$$

Moduli set $\{m_1, m_2, \dots, m_k\}$ defines a specific RNS. According to the Chinese Reminder Theorem (CRT) [16], such representation for any number X from a range $[0, M)$, where $M = m_1 \cdot m_2 \cdot \dots \cdot m_k$, are unique only if all m_i are pairwise co-prime, i.e. $\gcd(m_i, m_j) = 1$, for all $i \neq j,\ i,j = 1,2,\dots,k$. The number M is usually called the range of representation of numbers in RNS.

Since $\log_2 m_i \ll \log_2 M$, such a representation can be used in various areas. For example, RNS can be used for high-performance data processing using parallel computing facilities [32]. This property is useful for distributed storage of data dividing the data into practically equal parts.

To convert residues to the positional form of a number based on RNS code, the following consequence of CRT is used:

$$X = \left| \sum_{i=1}^{k} x_i \frac{M}{m_i} \left| \left(\frac{M}{m_i}\right)^{-1} \right|_{m_i} \right|_M ,$$

where $|t^{-1}|_{m_i}$ is the multiplicative inversion of some number t modulo m_i. In [35], a simplification of this approach was proposed. That approach based on the replacement of calculations by the large modulus M by calculations modulo 2^N, which are much simpler in software and hardware implementation. According to [35], $X = \tilde{X}M$ and

$$\tilde{X} = \left| \sum_{i=1}^{k} x_i b_i \right|_{2^N} ,$$

where $b_i = \left\lceil \frac{2^N}{m_i} \left| \left(\frac{M}{m_i}\right)^{-1} \right|_{m_i} \right\rceil$, $N = \lceil \log_2 M \sum_{(i=1)}^{k} (m_i - 1) \rceil - 1$ is the precomputed constants.

By adding to the moduli set $\{m_1, m_2, \dots, m_k\}$ redundant moduli $m_{k+1}, m_{k+2}, \dots, m_n$ and expanding the representation of the number $X \in [0, M)$ residues $x_{k+1}, x_{k+2}, \dots, x_n$ by division by new moduli, we obtain RRNS, which has new properties.

If $m_i < m_j$ for all m_i, $i = 1,2,\dots,k$, and m_j, $j = k+1, k+2, \dots, n$, then the loss of any $n-k$ does not violate the ability to restore the original number X. This allows us to achieve data availability in distributed storage. Note that, firstly, the moduli in the full base system $\{m_1, m_2, \dots m_n\}$ must be pairwise relatively co-prime and, secondly, X must belong to the interval $[0, M)$ and not belong to the interval $[M, M_n)$, where $M_n = M \cdot m_{k+1} \cdot \dots \cdot m_n$ is a full range of RRNS.

The most important feature of RRNS is the ability to control the integrity of information [17]. Subject to the conditions described above imposed on the full range of RRNS, it is possible to detect the presence of distortions in the $n-k$ residues of the number X represented in the RRNS [17, 20-21]. For this, it is necessary to restore the positional value of X according to the full moduli set of RRNS $\{m_1, m_2, \dots m_n\}$. If the obtained value $X^* \in [0, M)$, then the obtained value can be considered as a correct and assumed that $X = X^*$. Otherwise, if $X^* > M$, then in one or more of the residues there was a distortion. At the same time, different algorithms [17, 20-21] allow to localize the distortion, if it occurred in no more than $\lfloor (n-k)/2 \rfloor$ residues.

The properties described above make RRNS useful in the design of verifiable SSSs [36]. Our proposal combines all of the useful features of RRNS to design efficient secret sharing scheme with a high level of security and opportunity to control data integrity.

IV. Properties of Secret Sharing Schemes

Let $s \in S$ is some information (secret) that belongs to the secret space S. TSSS with (k, n) parameters means that information $s \in S$ (secret) is divided into n pieces (shares) $s_1, s_2, \dots, s_k, s_{k+1}, \dots, s_n$ so that it can be restored, if k or more shares are available, where $2 < k < n$. The number k is called the threshold of the scheme.

The set of shares with numbers from set $U = \{u_1, u_2, \dots, u_{k_1}\}$, where $u_i \in 1,2,\dots,n$, $u_i \neq u_j$, $i \neq j$, and $k_1 \geq k$, is called the authorized coalition (subset) of secret shares. There must be an algorithm for recovering the original secret s. While, the set $\widetilde{U} = \{u_1, u_2, \dots, u_{k_2}\}$, where $0 < k_2 < k$ is called an unauthorized coalition (subset).

Security of the SSSs is based on the impossibility in an acceptable time to recover the secret if there is any unauthorized coalition $\widetilde{U}$ of shares.

Perfect security is guaranteed by PSSS for which the following condition [30] is satisfied.

$$\Delta(s_u : u \in \widetilde{U}) = H(s \in S) - H(s \in S | s_u : u \in \widetilde{U}) = 0,$$

where $\Delta(Z)$ is a loss of entropy for the scheme for the set X of some possible shares of secret, $H(Z)$ is an informational entropy according to Shannon for the chosen set Z.

The expression above means that the information entropy of the secret space, in the presence of information about the secret shares belonging to the unauthorized coalition $\widetilde{U}$, must coincide with the information entropy of the given set in the absence of any information about the shares.

The classic PSSS is the Shamir scheme [12], which is actively used to store keys and other sensitive information. One of the most important SSS based on RNS is the asymptotically perfect Asmuth-Bloom (k, n) threshold scheme [30], where the secret must be chosen from the interval $[0, m_0)$. Moduli set is chosen in such a way that the conditions $m_0 < m_1 < \cdots < m_k < \cdots < m_n$ are performed and

$$\prod_{i=1}^{k} m_i > m_0 \prod_{i=0}^{k-2} m_{n-i}$$

To share a secret, a random number r is generated such that $s' = s + rm_0 < M$, where s an initial secret and $M = m_1 \cdot m_2 \cdot \ldots \cdot m_k$. The secret shares are calculated by the rule $s_i = s' \bmod m_i$, where $i = 1,2,\ldots,n$.

To recover the secret for any authorized coalition with the numbers from U, we use shares $s_{u_1}, s_{u_2}, \ldots, s_{u_{k_1}}$ as a number in the RNS with the moduli set $m_{u_1}, m_{u_2}, \ldots, m_{u_{k_1}}$, to obtain the positional value s' and the secret value using the rule $s = s' \bmod m_0$.

The asymptotic perfectness of the Asmuth-Bloom scheme means [30] that a sequence $m_0, m_1, m_2, \ldots, m_n$ can be chosen such that $\Delta(s_u : u \in \widetilde{U}) < \varepsilon$ for any $\varepsilon > 0$. According to [37], such a sequence must be compact. A compact sequence is a sequence of relatively prime numbers $m_0 < m_1 < \cdots < m_n$, such that $m_n < m_0 + m_0^{\theta}$ where $\theta \in (0,1)$. The safety of the Asmut-Bloom scheme with a compact sequence as a moduli set is based on the closeness of module sizes.

From the definition of perfect and asymptotically PSSS, it follows that such schemes require a significant redundancy. In practice, the requirement of perfectness means that the size of every share should be larger than the size of the secret. However, such a high level of security is often not required in real systems, while redundancy could be critical, for instance, in multi-cloud data storages.

According to [14], CS SSS is a scheme with the property of polynomial indistinguishability of inputs and outputs. This property is usually used to define the computational security of various cryptographic primitives [38].

A secret sharing scheme is computationally secure if it is impossible to establish the correspondence between shares collected by unauthorized coalitions $\widetilde{U}'$, $\widetilde{U}''$ and the secrets s', s'' obtained by that sets of shares, in polynomial time.

Such a scheme is the (k, n) threshold scheme by Krawczyk [14], which provides the sharing of the secret in several stages.

The secret s is encrypted by a strong secure symmetric-key data encryption scheme ENC_K with a secret key K. Obtained ciphertext $E = ENC_K(s)$ is divided into n parts $E_1, E_2, \ldots, E_n$ using Information Rabin IDA. The secret key is also divided into n parts $K_1, K_2, \ldots K_n$ based on some PSSS (for example, Shamir scheme). Hence, the shares of the secret are represented as $s_i = (E_i, K_i)$, $i = 1,2,\ldots,n$.

To recover the secret, it is enough to obtain any k shares and use the schemes of Rabin and Shamir to get the values E and K, then decrypt the original secret s.

In case of storage of a large amount of data in a multi-cloud system, the key can be generated once for a large data block and delivered for each storage separately. It allows reducing the redundancy in comparison with the PSSS.

V. A New Computationally Secure Verifiable Secret Sharing Based on RRNS

A. Distribution and reconstruction procedures

Rabin's Information Dispersal Algorithm can be classified as a complex algorithm. A significant simplification is the use of the RNS in place of the Rabin scheme. This section presents a formal description of the proposed scheme.

The RRNS moduli set $\{m_1, m_2, \ldots, m_k, m_{k+1}, \ldots, m_n\}$ must be chosen such that the sequence $m_1, m_2, \ldots, m_n$ is compact. Let $M = m_1 \cdot m_2 \cdot \ldots \cdot m_k$ and the secret s belong to the interval $S = [0, M)$. This can be achieved by dividing the original data into chunks of $\log_2 M$ bit in size. An important part of the scheme is a strong secure encryption algorithm ENC_K with the secret key K and the decryption algorithm DEC_K. For ENC_K, the size of the encrypted data must match the size of the source data. The key K is generated once for a large enough data block (for example, the file or the set of files).

Distribution:

1) encrypt the initial secret s based on the key K : $E = ENC_K(s)$;
2) disperse the obtained data E into n parts $E_1, E_2, \ldots, E_n$ according to rule $E_i = E \bmod m_i$;
3) share the key K into n parts $K_1, K_2, \ldots, K_n$ by using PSSS;
4) i-th share is $s_i = (E_i, K_i)$ for all $i = 1,2,\ldots,n$.

If $K < M$, in stage 3 of PSSS, we can choose the Asmut-Bloom scheme with $m_0 = M$. Suppose further that there is some authorized coalition of shares with numbers from $U = \{u_1, u_2, \ldots, u_{k_1}\}$ where $k \leq k_1 \leq n$.

Reconstruction (variant 1):

1) reconstruct the key K based on a combination of key's parts $K_{u_1}, K_{u_2}, \ldots, K_{u_{k_1}}$;
2) reconstruct the value E based on residues $E_{u_1}, E_{u_2}, \ldots, E_{u_{k_1}}$ by conversion from RNS representation into binary form;
3) reconstruct the initial secret s by the rule $s = DEC_K(E)$.

A feature of the proposed scheme is the ability to verify the secret due to the corrective properties of the RRNS. Known algorithms for correcting errors in the RNS code can be used for this [17, 20-21].

It allows detecting distortions in $n-k$ shares and localizing distortions in $\lfloor(n-k)/2\rfloor$ residues.

To apply these features, the following version of the secret reconstruction procedure is suggested. We assume that an algorithm $L = LOC(E_1, E_2, \dots, E_n)$, where $L = \{l_1, l_2, \dots, l_h\}$ is the numbers of undistorted residues, $k \le h \le n$, $1 \le l_i \le n$, $l_i \ne l_j$ when $i \ne j$, is used to localize distortions. Suppose we have n shares.

Reconstruction (variant 2, verifiable):

1) reconstruct the key K based on a combination of key's parts $K_1, K_2, \dots, K_n$;
2) reconstruct the value E^* based on residues $E_1, E_2, \dots, E_n$ by conversion from RNS representation into binary form and
    a) if $E^* < M$ then there is no distortion in the stored data, assume that $E = E^*$ and go to stage 3,
    b) if $E^* \ge M$ then made error localization $L = LOC(E_1, E_2, \dots, E_n)$ and the value E reconstructed by residues $E_{l_1}, E_{l_2}, \dots, E_{l_h}$, which do not contain the detected distorted residues;
3) reconstruct the initial secret s by the rule $s = DEC_K(E)$.

In case the key K is shared using the Asmut-Bloom scheme the error detection and localization procedure similar to the stage 2 can be applied to provide the integrity of the secret key. It should only be taken into account that for the Asmuth-Bloom scheme, a different moduli set where each modulus has a larger value should be used.

B. Computational security of the proposed scheme

In this section, we will prove the computational security of the proposed SSS. Let some secret s has been shared by the method described in the previous subsection and an adversary has collected some unauthorized coalition of shares with numbers from $\widetilde{U}$. Based on this coalition the adversary can recreate the encrypted secret only partially by combining the shares with numbers from $\widetilde{U}$. In this case, he obtains the value $\tilde{E}$ determined by the next expression

$$\tilde{E} = E \bmod \widetilde{M},$$

where $\widetilde{M} = \sum_{i \in \widetilde{U}} m_i$. However, this is not enough to get the value E. From the expression above, it follows that $E = \alpha\widetilde{M} + \tilde{E}$. To accurately determine E, it is necessary to find out the unknown value α. Since the values E has a uniform distribution over all possible values of the secret obtained after perfectly secure encryption, value α with an uniform probability may be an any value from the interval

$$-\frac{\tilde{E}}{\widetilde{M}} \le \alpha < \frac{M - \tilde{E}}{\widetilde{M}}$$

This, in fact, means that adversary has to test exactly $\tau = \left\lfloor \frac{M}{\widetilde{M}} \right\rfloor + 1$ possible values of α. It can be experimentally determined that τ tends to m_1 for the worst case of an unauthorized coalition $\widetilde{U}$, if the sequence $m_1 < m_2 < \dots < m_n$ is compact.

However, by such a brute force attack, it is possible to find just a value E, which is the encrypted part of the secret. The encryption key is shared using the perfect secret sharing scheme. According to [14], this means that, even if it is possible to truncate the space for brute force attack to recover E and distinguish the values E', E'' with residues $E'_{u_1}, E'_{u_2}, \dots, E'_{u_{k_2}}$ and $E''_{u_1}, E''_{u_2}, \dots, E''_{u_{k_2}}$, where $u_i \in \widetilde{U}$, it is not possible to distinguish the secrets s' and s'' without hacking the encryption scheme ENC or reconstructing the key K based on the incomplete set of shares $K_{u_1}, K_{u_2}, \dots, K_{u_{k_2}}$.

This proves that selecting perfect secure encryption scheme ENC and the PSSS for key sharing is important.

It should be noted that it is expediently to generate a new key K for a large data block (for example, a file) in order to provide a high complexity to key breaking algorithm on all stored data.

To maintain a high cryptographic security level of the secret sharing and the encryption scheme ENC, the value of key must be large enough. It affects the selection of the RRNS parameters and PSSS. The same requirements are applied to the scheme by Krawczyk [14].

VI. Performance Analysis

The main computational process of the proposed scheme requires significant computing resources. It is the dispersal of the encrypted secret E using RNS.

In multi-cloud storage, the process of data distribution and reconstruction can be repeated many times. To assure the cryptographic security requirements, the size of the shares and, therefore, the size of the secret must be large enough. In this paper, we consider that the size of a single share is up to 256 bits.

The classical Rabin IDA requires calculations in the Galois field $GF(2^w)$ that is the basis for many forward error correction schemes, where w is the share size.

It leads to the operations on polynomials constructed over the field $GF(2)$ and modulus, which is equal to an irreducible polynomial in this field.

To distribute and recover data in the Rabin scheme, it is necessary to multiply the vector of k pieces of input data s by a pre-generated matrix of size $n \times k$ over the Galois field $GF(2^w)$.

One of the inverse submatrices is multiplied by a vector from the available k data parts to restore the original value of s. Usually, the Cauchy and Vandermonde matrices are used. Matrix-vector multiplication requires $n \cdot k$ multiplications and $n \cdot (k-1)$ additions, which are polynomial or modular operations depending on w.

Such an operation entails significant computational costs. Its computational complexity is $O(n \cdot k \cdot w \cdot \log w \log\log w)$ of bitwise operations.

In contrast to the Rabin scheme, the proposed scheme is based on RNS. For RNS, the operations of distribution and reconstruction of data are reduced to the conversion of the number into the RNS and to the reverse conversion into the binary representation, respectively.

Forward RNS conversion leads to calculating the remainder of the division of $s \in [0, M)$ by moduli $m_1, m_2, \dots, m_n$. The reverse conversion can be performed according to the simplified equation presented in Section III.

Let w is a smallest value such that $m_n < 2^w$. Then, the computational complexity of data encoding of forward RNS conversion is equal to $O(n \cdot k \cdot w)$.

To confirm the computational simplicity of the RNS in comparison with the Rabin scheme, a complex of Java classes are developed. The experiments are performed on a PC with Windows 10, Intel Core i5 4200U for JRE 1.8.0_121. Irreducible polynomials over a field of modulo 2 and RNS moduli sets are selected for shares sizes from 32 to 256 bits (Table II). We study data distribution and data recovering for (4,4) TSSS based on both approaches.

TABLE II. IRREDUCIBLE POLYNOMIALS FOR RABIN SCHEME AND RNS MODULI SETS WHICH WERE USED FOR MODELING

Share sizes w, *bit*	**Irreducible polynomials for Rabin scheme**	**RNS moduli sets**
		$\{m_1, m_2, m_3, m_4\}$
32	$x^{32} + x^7 + x^3 + x^2 + 1$	$\{2^{32}, 2^{32} + 1, 2^{32} + 3, 2^{32} + 5\}$
64	$x^{64} + x^4 + x^3 + x + 1$	$\{2^{64}, 2^{64} + 1, 2^{64} + 3, 2^{64} + 5\}$
96	$x^{96} + x^6 + x^5 + x^3 + x^2 + x + 1$	$\{2^{96}, 2^{96} + 1, 2^{96} + 3, 2^{96} + 5\}$
128	$x^{128} + x^7 + x^2 + x + 1$	$\{2^{128}, 2^{128} + 1, 2^{128} + 3, 2^{128} + 5\}$
160	$x^{160} + x^5 + x^3 + x^2 + 1$	$\{2^{160}, 2^{160} + 1, 2^{160} + 3, 2^{160} + 5\}$
192	$x^{192} + x^7 + x^2 + x + 1$	$\{2^{192}, 2^{192} + 1, 2^{192} + 3, 2^{192} + 5\}$
224	$x^{224} + x^8 + x^7 + x^5 + x^4 + x^2 + 1$	$\{2^{224}, 2^{224} + 1, 2^{224} + 3, 2^{224} + 5\}$
256	$x^{256} + x^{10} + x^5 + x^2 + 1$	$\{2^{256}, 2^{256} + 1, 2^{256} + 3, 2^{256} + 5\}$

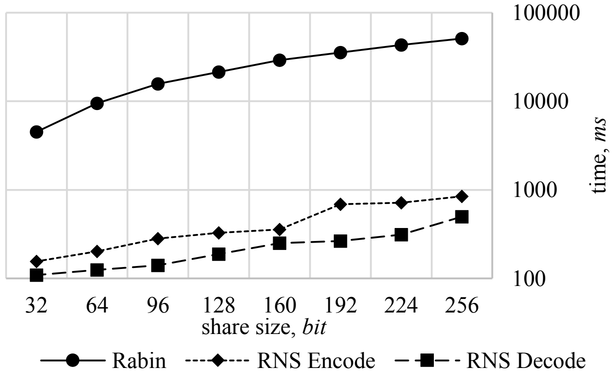

Figure 1. Simulation results for Rabin scheme encoding and RNS encoding/decoding phases

We estimate the speed of performing data distribution operations according to the Rabin scheme (which corresponds to the reconstruction operation) and operations of encoding and decoding data in RNS.

Each iteration is fast in both algorithms. Therefore, for each share size, 100 000 iterations are performed to obtain more reliable results (Figure 1).

Figure 1 shows that encoding and decoding of data in RNS are much faster in comparison with encoding (or decoding) by the Rabin scheme.

Both phases of Rabin scheme are the same and performed as the vector-matrix multiplication. Encoding time for Rabin scheme is increased more smoothly in comparison with RNS encoding/decoding. It is because the complexity of Rabin scheme depends only on the size of shares.

On the other hand, the complexity of RNS decoding/encoding is depended also on the chosen moduli set. Therefore, moduli have an impact on the speed of computation, which is notable the Figure 1.

Obtained results coincide with our theoretical estimation of the algorithm complexity. Difference between computational complexity of encoding/decoding phases of Rabin scheme $O(n \cdot k \cdot w \cdot \log w \log\log w)$ and RNS encoding phase $O(n \cdot k \cdot w)$ has the same order that the difference between the execution times obtained during simulations.

VII. CONCLUSION

We propose and analyze a new computationally secure secret sharing scheme based on RNS. We show that our solution is more efficient for data distributing than the Rabin scheme. It avoids complex matrix operations in a polynomial finite field. Moreover, it includes the approaches to data integrity control based on the RRNS code. Our scheme is the combination of RNS encoding with encryption to obtain a new property of secret sharing based on RNS. The major advantage over conventional SSSs and SSSs based on RNS is low data redundancy. It is achieved by reducing the security level within acceptable limits.

One of the main contributions of our work is a strict proof of computational security of the proposed scheme. An adversary cannot reconstruct the secret with less than k shares. Moreover, our solution can tolerate up to k faulty shareholders.

These properties make the new CS SSS applicable for multi-cloud storage systems. It is verifiable and able to confirm the correctness of the decoded secret. A distinctive feature of the scheme is the using RRNS properties for verification. This approach reduces the verification time in comparison with the existing ones by reducing the number of operations with data.

This scheme maintains confidentiality, integrity, and availability of data. It combines advantages of computationally secure secret sharing schemes with high-speed data processing and verifiability.

ACKNOWLEDGMENT

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

REFERENCES

[1] H. Abu-Libdeh, L. Princehouse and H. Weatherspoon, “RACS: a case for cloud storage diversity,” in Proceedings of the 1st ACM symposium on Cloud computing, 2010, pp. 229–240.

[2] A. Bessani, M. Correia, B. Quaresma, F. André and P. Sousa, “DepSky: dependable and secure storage in a cloud-of-clouds,” ACM T. Storage, vol. 9, no. 4, 2013, pp. 12:1–12:33.

[3] N. Chervyakov, M. Babenko, A. Tchernykh, N. Kucherov, V. Miranda-López and J.M. Cortés-Mendoza, “AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security,” Future Gener. Comp. Sy., 2017, in press.

[4] A. C. Mora, Y. Chen, A. Fuchs, A. Lane, R. Lu, P. Manadhata et al. “Top ten big data security and privacy challenges,” Cloud Security Alliance. https://www.isaca.org/Groups/Professional-English/big-data/GroupDocuments/Big_Data_Top_Ten_v1.pdf, 2012 (accessed 21.06.17).

[5] L. Rizzo, “Effective erasure codes for reliable computer communication protocols,” in ACM SIGCOMM Computer Communication Review, vol. 27, no. 2, 1997, pp. 24–36.

[6] M.A. AlZain, B. Soh and E. Pardede, “A survey on data security issues in cloud computing: From single to multi-clouds”, in IEEE System Science (HICSS), 2012 45th Hawaii International Conference on, 2012, pp. 5490–5499.

[7] A. Tchernykh, U. Schwiegelsohn, E.G. Talbi and M. Babenko, “Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability,” J. Comput. Sci.-Neth., 2016, in press.

[8] B. Fabian, T. Ermakova and P. Junghanns, “Collaborative and secure sharing of healthcare data in multi-clouds,” Inform. Syst., vol. 48, 2015, pp. 132–150.

[9] K. Shvachko, H. Kuang, S. Radia and R. Chansler “The Hadoop distributed file system,” in IEEE 26th symposium on Mass storage systems and technologies (MSST), 2010, pp. 1–10.

[10] S. Ghemawat, H. Gobioff and S.T. Leung, “The Google file system,” in 19th Symposium on Operating Systems Principles, 2003.

[11] P. Junghanns, B. Fabian and T. Ermakova, “Engineering of secure multi-cloud storage,” Comput. Ind., vol. 83, 2016, pp. 108–120.

[12] A. Shamir, “How to share a secret,” Commun. ACM, vol. 22, no. 11, 1979, pp. 612-613.

[13] J.M. Bohli, N. Gruschka, M. Jensen, L.L. Iacono and N. Marnau, “Security and privacy-enhancing multicloud architectures,” IEEE T. Depend. Secure, vol. 10, no. 4, 2013, pp. 212–224.

[14] H. Krawczyk, “Secret sharing made short,” in Springer Annual International Cryptology Conference, 1993, pp. 136-146.

[15] J.J. Wylie, M.W. Bigrigg, J.D. Strunk, G.R. Ganger, H. Kiliccote and P.K. Khosla, “Survivable information storage systems,” Computer, vol. 33, no. 8, 2000, pp. 61–68.

[16] N.S., Szabo and R.I. Tanaka, “Residue arithmetic and its applications to computer technology,” McGraw-Hill, 1967.

[17] C. Ding, D. Pei and A. Salomaa, “Chinese remainder theorem: applications in computing, coding, cryptography,” World Scientific, 1996.

[18] C. Asmuth and J. Bloom, “A modular approach to key safeguarding,” IEEE T Inform Theory, vol. 29, no. 2, 1983, pp. 208–210.

[19] M. Mignotte “How to share a secret,” in Springer Workshop on Cryptography, 1982. pp. 371–375.

[20] T.F. Tay and C.H. Chang, “Fault-tolerant computing in redundant residue number system,” in Embedded Systems Design with Special Arithmetic and Number Systems, Springer, Cham, 2017, pp. 65–88.

[21] V.T. Goh and M.U. Siddiqi, “Multiple error detection and correction based on redundant residue number systems,” IEEE T. Commun., vol. 56., no 3., 2008, pp. 325–330.

[22] J. Li and B. Li, “Erasure coding for cloud storage systems: a survey,” Tsinghua Sci. Technol., vol. 18, no. 3, 2013, pp. 259–272.

[23] S.B. Wicker, “Error control systems for digital communication and storage, ” vol. 1, Englewood Cliffs: Prentice hall, 1995.

[24] M.O. Rabin, “Efficient dispersal of information for security, load balancing, and fault tolerance,” J. ACM, vol. 36, no. 2, 1989, pp. 335–348.

[25] H. Krawczyk, “Distributed fingerprints and secure information dispersal,” in Proceedings of the twelfth annual ACM symposium on Principles of distributed computing, 1993, pp. 207-218.

[26] S.B. Wicker and V.K. Bhargava, “Reed-Solomon codes and their applications,” John Wiley & Sons, 1999.

[27] V. Attasena, J. Darmont and N. Harbi, “Secret sharing for cloud data security: a survey,” The VLDB Journal, vol. 26 ,no. 5, 2017, pp. 657–681.

[28] A. Beimel, “Secret-sharing schemes: a survey,” in International Conference on Coding and Cryptology, 2011, pp. 11–46.

[29] M. Stadler, “Publicly verifiable secret sharing,” in Advances in Cryptology - EUROCRYPT '96, Lecture Notes in Computer Science, vol. 1070, 1996, pp. 190–199.

[30] M. Quisquater, B. Preneel and J. Vandewalle, “On the security of the threshold scheme based on the Chinese remainder theorem,” in International Workshop on Public Key Cryptography, 2002, pp. 199–210.

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

[32] C.H. Chang, A.S. Molahosseini, A.A.E. Zarandi and 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., vol. 15, no. 4, 2015, pp. 26–44.

[33] 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, 2016, pp. 208-218.

[34] M. Gomathisankaran, A. Tyagi and K. Namuduri, “HORNS: A homomorphic encryption scheme for Cloud Computing using Residue Number System.m” in IEEE 45th Annual Conference on Information Sciences and Systems, 2011.

[35] N.I. Chervyakov, A.S. Molahosseini, P.A. Lyakhov, M.G. Babenko and M.A. Deryabin, “Residue-to-binary conversion for general moduli sets based on approximate Chinese remainder theorem,” Int. J. Comput. Math., vol. 94, no. 9, 2017, pp. 1833–1849.

[36] K. Kaya and A.A. Selçuk, “A verifiable secret sharing scheme based on the chinese remainder theorem,” in International Conference on Cryptology in India, 2008, pp. 414–425.

[37] M. Barzu, F.L. Ţiplea and C.C. Drăgan, “Compact sequences of co-primes and their applications to the security of CRT-based threshold schemes,” Inform. Sciences, vol. 240, 2013, pp. 161–172.

[38] O. Goldreich, “A note on computational indistinguishability,” Inform. Process. Lett.,” vol. 34, no. 6, 1990, pp. 277–281.