Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security

Nikolay Chervyakov [a], Mikhail Babenko [a], Andrei Tchernykh [b,*], Nikolay Kucherov [a], Vanessa Miranda-López [b], Jorge M. Cortés-Mendoza [b]

[a] North-Caucasus Federal University, Stavropol, Russia
[b] CICESE Research Center, Ensenada, BC, Mexico

HIGHLIGHTS

- Introduce error detection, correction, and controlling computational results schemes based on residue number system.
- Introduce the concept of an approximate value of a rank of a RNS number.
- Discuss the ways to reduce risks of information loss, denial of access, interruptions in communication and information leakage.
- Study overheads of data encoding and decoding.
- Provide theoretical basis to configure storage properties to the demanded level of reliability, scalability, and performance.

ARTICLE INFO

Article history:
Received 9 April 2017
Received in revised form 19 August 2017
Accepted 22 September 2017
Available online xxxx

Keywords:
Big data storage
Multi-cloud
Internet of Things
Resource management
Security
Safety
Reliability
Residue number system

ABSTRACT

Benefits of Internet of Things and cloud–fog-edge computing are associated with the risks of confidentiality, integrity, and availability related with the loss of information, denial of access for a long time, information leakage, conspiracy and technical failures. In this article, we propose a configurable, reliable, and confidential distributed data storage scheme with the ability to process encrypted data and control results of computations. Our system utilizes Redundant Residue Number System (RRNS) with new method of error correction codes and secret sharing schemes. We introduce the concept of an approximate value of a rank of a number (AR), which allows us to reduce the computational complexity of the decoding from RNS to binary representation, and size of the coefficients. Based on the properties of the approximate value and arithmetic properties of RNS, we introduce AR-RRNS method for error detection, correction, and controlling computational results. We provide a theoretical basis to configure probability of information loss, data redundancy, speed of encoding and decoding to cope with different objective preferences, workloads, and storage properties. Theoretical analysis shows that by appropriate selection of RRNS parameters, the proposed scheme allows not only increasing safety, reliability, and reducing an overhead of data storage, but also processing of encrypted data.

1. Introduction

According to the IT company "Industrial Development Corporation", the total amount of data in the world has increased by nine times within five years (Gantz & Reinsel, 2011) [1]. This number is expected to double at least every two years.

A number of challenges arise from data capture and data processing. Novel techniques and technologies to excavate and store data to benefit our specified purposes are being created and developed. Data-intensive science, especially, data-intensive computing, Internet of Things (IoT), edge computing, cyber–physical systems, etc., is coming into play providing tools to aggregate, store, and process data (Ahmed et al., 2017) [2].

Over the last decade, there has been considerable interest to challenge this tendency by using Residue Number System (RNS) (Chang et al., 2015) [3].

Big data is defined as high-volume, high-velocity, and/or high-variety information assets that require new forms of processing to enhance decision making, insight discovery, and storage optimization.

Addressing big data is a challenging task required a large computational infrastructure to ensure successful data storage,

* Corresponding author.
E-mail addresses: ncherviakov@ncfu.ru (N. Chervyakov), mgbabenko@ncfu.ru (M. Babenko), chernykh@cicese.mx, chernykh@cicese.edu.mx (A. Tchernykh), nkucherov@ncfu.ru (N. Kucherov), vmiranda@cicese.edu.mx (V. Miranda-López), jcortes@cicese.edu.mx (J.M. Cortés-Mendoza).

https://doi.org/10.1016/j.future.2017.09.061
0167-739X/

processing and analysis. Nowadays, well-known cloud storage providers, such as Dropbox, Google Drive, Copy, AmazonS3, SkyDrive, etc., are widely used in various spheres of human life to address these challenges.

IoT provides technologies that deliver connectivity mechanisms to IoT devices, mobile and cloud-based applications, and generate data to be stored, analyzed, and act based on it.

The design of distributed big data storage systems has to take into account that data are numerous, cannot be categorized into regular relational databases, and should be captured and processed rapidly.

Traditional systems store data in structured Relational DataBase Management Systems (RDBMS), Hadoop File Systems (Shvachko et al., 2010) [4], replication (Ghemawat et al., 2003) [5], etc.

Intensive and extensive studies examine different aspects of cloud data storages. However, mitigating risks of confidentiality, integrity, availability, etc. has not been adequately addressed in the scientific literature.

Zhang et al., 2013 [6] discussed the problem of preserving the privacy of intermediate datasets in cloud computing. The authors argued that encrypting all intermediate datasets in the cloud is neither computationally effective nor cost effective because an encryption and decryption of data take a long time.

The Cloud Security Alliance (CSA) presented top 10 challenges of data security and privacy (Mora et al., 2012) [7]. CSA detects numerous deliberate and accidental threats (Hubbard and Sutton, 2010) [8].

Deliberate threats include unauthorized access to the information, interception, falsification, forgery, hacker attacks, etc. CSA states that, in recent years, the number of unauthorized accesses to the information processed and stored in the clouds is dramatically increased. Cryptographic protocols and error correction codes can be used to reduce this risk. However, the use of classical symmetric and asymmetric ciphers requires large computational power and is not applicable to mobile devices (Singh et al., 2017) [9].

Accidental threats include user errors, carelessness, curiosity, etc. The information control and protection system based on the proactive concept can be used. The proactive concept includes the simultaneous use of weighted secret sharing scheme based on RRNS, encryption keys, and checksums for monitoring obtained results.

An alternative way to ensure information confidentiality is to use homomorphic encryption based on RNS (Cheon et al., 2015) [10], (Tchernykh et al., 2016) [11].

Performance and scalability are two important factors on big data processing. The storage infrastructure has to provide reliable storage space with powerful access interface for query and analysis (Lynch 2008) [12].

Distributed storage can be based on multiple clouds. Usually, data is divided into multiple pieces to be stored at different clouds to ensure availability in case of failure. However, failures of distributed storage may cause inconsistency among different copies of the same data (Ghemawat et al., 2003) [5].

One can use large databases. In this case, for high performance, data processing and analysis have to be carried out by parallel computing (Fernández et al., 2014) [13].

Chen and Huang 2013 [14] proposed a modified framework of MapReduce using full homomorphic encryption. The main drawbacks of this framework are data redundancy, computational complexity of data encryption algorithms, and low reliability. To eliminate these drawbacks, (Celesti et al., 2016) [15] proposed a reliable cloud storage system based on RNS.

In this paper, we focus on configurable and reliable RRNS systems in multi-cloud environments to ensure security, robustness, confidentiality, and efficient functionality.

The most resource-consuming operation in the implementation of RNS to binary conversion is the operation of finding RNS residue from dynamic range. To increase efficiency of data processing and decrease energy consumption of IoT devices during data encoding and decoding, we use RNS modules of a special form $2^b \pm \alpha$, which allow to find a residue of division with linear complexity.

We propose Approximation of the Rank (AR) that allows to substitute operations of finding residue by taking higher bits of a number based on the introduced function of computing the approximate rank of a RNS number. Based on the properties of the approximate value and arithmetic properties of RNS, we introduce AR-RRNS method for error detection, correction, and controlling computational results.

This paper is organized as follows. We review distributed storage systems for IoT-clouds in Section 2. In Section 3, we describe the main model of data storage. Section 4 presents configurable parameters used in our method. Section 5 introduces the rank of RNS number and its approximation. Section 6 focuses on error detection, correction and control of arithmetic operations, data processing and transfer protocol for wireless networks in IoT. Section 7 describes our configurable model. The conclusions and future work are discussed in the last Section 8.

2. Related work

In this section, we discuss the distributed data management technologies.

2.1. Distributed storage system

A variety of approaches can be used to construct a distributed system for storage and processing. Several of them are based on the cloud and grid computing paradigms (Vouk 2008) [16]. These infrastructures have common characteristics, but also principal differences.

The use of clouds for data storing requires a number of factors such as security, reliability and scalability under limited Internet connection bandwidth (Mora et al., 2012) [7], (Hubbard & Sutton, 2010) [8], (Ahmed & Rehmani, 2017) [17].

In order to provide quick access to distributed data and ensure a high degree of reliability, availability and scalability, (Chang et al., 2008) [18] proposed Bigtable system based on replication of not encrypted data, without providing privacy and data security.

An alternative mechanism is Hadoop and MapReduce based on splitting the dataset into independent chunks that are processed in parallel and reducing them (Dean & Ghemawat, 2008) [19]. However, as shown by (Herodotou et al., 2011) [20], its main drawback is the low efficiency.

Not relational databases (NoSQL) that take into account heterogeneity of unstructured data become popular (Leavitt, 2010) [21]. However, two most popular NoSQL databases, Cassandra and MongoDB, have problems with security and privacy of data (Okman et al., 2011) [22].

To solve these problems, the classic data encryption can be used, but it is not applicable in the case of data processing. It does not allow intensive analysis and processing of the data, and takes significant computing resources to perform the encryption and decryption of data.

An alternative solution is the homomorphic encryption introduced by (Rivest et al., 1978) [23] that allows encrypted data processing. Significant progress in the field of homomorphic encryption was achieved after the publication of (Gentry, 2009) [24], in which the fully homomorphic encryption is proposed. Classic fully homomorphic encryption built on ideal lattices leads to a large redundancy of stored data, which makes it inapplicable to big data storage systems. Other homomorphic encryption method is based on RNS that enables computational security and reliability (Section 2.4).

Please cite this article in press as: N. Chervyakov, et al., AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security, Future Generation Computer Systems (2017), https://doi.org/10.1016/j.future.2017.09.061.

2.2. Applications in distributed environments

There is wide range of approaches that are used for distributed data intensive computing (Venugopal et al., 2006) [25].

Distributed Data Base (DDB) stores data on various sites of a computer network and uses logics to organize the set of data (Ozsu & Valduriez, 1991) [26]. There are two ways to construct DDBs. Top-down approach takes a database and distributes it over various sites. While bottom-up approach unites number of distinct databases with one interface. The main field of application of DDBs is structured data storage, therefore, it is not applicable to arbitrary datasets, such as Big data.

Content Delivery Network (CDN) (Dilley et al., 2002) [27] is a set of (non-origin) servers that cache the data, satisfy the client requests to the database, and reduce the workload of origin servers. We can state the following principles of CDN: load balancing, bandwidth conservation and time efficiency. However, CDNs are not widely used in practice due to the fact that they are not flexible.

The main principles of P2P Network (Oram, 2001) [28] are scalability and reliability achieved by decentralized structure and redundancy, resources sharing and anonymity. P2P networks are efficient in providing fast access to files to a group of peers. Nevertheless, most P2P networks do not allow integrated computations and serve as data distribution environment.

2.3. Approaches for reliability and confidentiality

When clouds are used for data storage, reliability, scalability, security, confidentiality, and data processing in the encrypted form should be taken into consideration. These characteristics are also crucial for mobile devices, where technical characteristics and energy consumption are limited (Ahmad et al., 2017) [29].

Tchernykh et al., 2016 [9] show that distributed data storage under the conditions of uncertainty in cloud computing can use data replication, secret sharing schemes, Redundant Residue Number System (RRNS), erasure codes, regenerating codes, and homomorphic encryption.

Below, we describe main known methods of organizing distributed data storage in clouds and grids (Table 1) by comparing the following properties: reliability, scalability, availability, confidentiality, integrity, privacy, and homomorphic encryptions. These properties are briefly described in Appendix. L denotes the data size.

The most effective in terms of complexity is the method from work (Ghemawat et al., 2003) [4]. However, its main disadvantage is that the data is stored in an unencrypted form, which leads to a limited applicability.

An alternative approach to build a reliable storage system is to use error correction codes based on RRNS, Erasure codes (Dimakis et al., 2010) [35], and Regenerating codes (Lin et al., 2014) [52]. However, Erasure and Regenerating codes do not allow to process the encrypted data. For data processing, homomorphism is an important property of the system, since it allows processing encrypted data without additional computational cost for data decoding (Rivest et al., 1978) [23].

A significant breakthrough in the field of homomorphic encryption has occurred due to the work of (Gentry 2010 [37]), which proposed the fully homomorphic encryption scheme for performing both addition and multiplication with the cipher text. The main disadvantages of the proposed algorithm is a significant data redundancy and lack of tools to control the results of arithmetic operations.

In Table 1 particular attention deserves distributed data storage of (Gomathisankaran et al., 2011) [38], which assures the safety, confidentiality, homomorphism, reliability, and scalability of data. The authors proposed two approaches to construct systems based on homomorphic secret sharing schemes in RRNS. RRNS moduli are secret keys stored by a user. Data processing leads to exponential increase in the load of the network and memory, and makes it inapplicable in practice.

Secret sharing schemes proposed by (Asmuth & Bloom, 1983) [53] and (Mignotte 1982) [54] ensure the security and confidentiality of data. RRNS has similar properties to the Mignotte secret sharing scheme. Its arithmetic properties allow controlling results of data processing.

2.4. Redundant residue number system

RRNS represents original numbers as residues with respect to a moduli set. Thus, the number is split into smaller numbers, which are independent.

Let $p_1, p_2, \ldots, p_n$ are pairwise coprime numbers used as moduli set of RRNS, and $n = k+r$. Then RRNS range is defined $P = \prod_{i=1}^{k} p_i$.

Data is integer number X, where $X \in [0, P-1)$. X is defined in RRNS as a tuple $X \overset{RRNS}{\rightarrow} (x_1, x_2, \ldots, x_n)$, where $x_i = |X|_{p_i}$ represents the remainder of division of X by p_i.

In RRNS settings (k, n), using data from any k remainders 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 $r-1$ errors. For error isolation and correction, projection methods are used, where the number of calculated projections grows exponentially depending on the r value. As a result, RRNS is impractical without significant optimization.

Celesti et al., 2016 [15] proposed the use of RRNS for reliable and scalable cloud storage systems. Operations can be accomplished on them separately and concurrently, which makes the computations simpler and faster. Redundancy of residues allows to build system with multiple error detection and correction.

It also allows to perform arithmetic operations with the following property.

$$X * Y = (x_1, x_2, \ldots, x_n) * (y_1, y_2, \ldots, y_n) = \left(|x_1 * y_1|_{p_1}, |x_2 * y_2|_{p_2}, \ldots, |x_n * y_n|_{p_n}\right) \quad (1)$$

where $*$ denotes one of the operations: addition, multiplication, and subtraction. It holds: $x_i = |X|_{p_i}, y_i = |Y|_{p_i}$, for all $i = \overline{1, n}$.

Since representation of numbers in RRNS can be seen as secret sharing scheme, we can obtain computationally secure data storage. Based on RRNS property Eq. (1), we can state that the proposed system is a homomorphic cypher.

Gomathisankaran et al., 2011 [38] studied fully homomorphic cypher systems based on secret sharing in RNS. However, it should be noted that it is not practical to use RNS moduli set as the secret keys. It leads to high redundancy and resource intensive decoding that can be more complex than the original problem. As a consequence, we assume that RRNS moduli set can be used as a public key.

Cheon et al., 2015 [10] offered an alternative way of constructing homomorphic encryption system in RNS. They proposed a generalization of DGHV (Dijk, Gentry, Halevi, and Vaikuntanathan) algorithm, which allows to improve characteristics of computational complexity and redundancy. This scheme is based on the ideas of the secret sharing scheme in RRNS of (Asmuth & Bloom, 1983) [53]. The proposed algorithm has a high redundancy, compared with schemes in the classic RRNS.

Matutino et al., 2014 [55] presented efficient methods that allow to encode the data with minimal computational costs. Moduli set of the form $2^{b_i} \mp \alpha_i$ is used to efficiently perform scaling and arithmetic operations over encoded data.

In order to determine the problems in the data storage and data processing, we use properties of error detection and correction in RNS considered by (Barsi & Maestrini, 1974) [56]. Modification

Please cite this article in press as: N. Chervyakov, et al., AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security, Future Generation Computer Systems (2017), https://doi.org/10.1016/j.future.2017.09.061.

Table 1
Related work summary.

Method	Properties								
	Availability	Confidentiality	Homomorphic	Integrity	Privacy	Reliability	Scalability	Encryption	Decryption
Abu-Libdeh et al., 2010 [30]	•					•		$O(L \log L)$	$O(L \log L)$
Adya et al., 2002 [31]	•	•		•	•	•	•	$O(L^2)$	$O(L^2)$
Ateniese et al., 2006 [32]		•		•	•	•		$O(L)$	$O(L^2)$
Bessani et al., 2013 [33]	•	•		•	•	•		$O(L)$	$O(L)$
Bowers et al., 2009 [34]	•			•	•	•	•	$O(L^2)$	$O(L^2)$
Celesti et al., 2016 [15]	•			•	•	•		$O(L^2)$	$O(L^2)$
Dimakis et al., 2010 [35]		•		•	•	•	•	$O(L^2)$	$O(L^2)$
Erkin et al., 2012 [36]			•		•		•	$O(L^2)$	$O(L^2)$
Gentry, 2010 [37]			•		•			$O(L^6 \log L)$	$O(L^2 \log L)$
Ghemawat et al., 2003 [5]	•	•				•	•	$O(1)$	$O(1)$
Gomathisankaran et al., 2011 [38]		•	•	•	•	•	•	$O(L^2)$	$O(L^2)$
Kong et al., 2010 [39]		•		•	•	•	•	$O(L \log L)$	$O(L \log L)$
Li et al., 2010 [40]	•	•		•	•	•	•	$O(L^2)$	$O(L^2)$
Lin&Tzeng 2012 [41]	•	•		•	•	•	•	$O(L \log L)$	$O(L \log L)$
Pang&Wang 2005 [42]	•	•		•	•	•	•	$O(L^2)$	$O(L^2)$
Parakh&Kak2011 [43]	•	•		•	•	•	•	$O(L)$	$O(L \log^2 L)$
Parakh&Kak2009 [44]		•		•	•	•	•	$O(L)$	$O(L \log^2 L)$
Ruj et al., 2011 [45]	•	•		•	•			$O(L^2)$	$O(L^2)$
Samanthula et al., 2015 [46]			•		•			$O(L^2)$	$O(L^2)$
Sathiamoorthy et al., 2013 [47]		•				•	•	$O(L \log^2 L)$	$O(L \log^2 L)$
Shah et al., 2012 [48]		•				•	•	$O(L)$	$O(L)$
Wang et al., 2012 [49]	•	•		•	•	•	•	$O(L^3)$	$O(L^3)$
Wylie et al., 2000 [50]	•	•		•			•	$O(L^3)$	$O(L^3)$
Yang et al., 2004 [51]	•	•		•			•	$O(L^2)$	$O(L^2)$

and improvements of detection and error correction in the RRNS are considered in (Goh & Siddiqi, 2008) [57], (Chervyakov et al., 2016) [58], etc.

The common issue for the majority of the proposed works is to detect and correct one error. When reliability is provided for a single computer, the detection and correction of a single error is sufficient. However, when we consider big data, it is necessary to have efficient algorithms for detecting and correcting several errors.

The RRNS scheme for data storage provides a secure, reliable and scalable storage. It has properties of error correction codes, and two cryptographic primitives: secret sharing schemes and homomorphic encryption, which makes it useful for data processing in the encrypted form.

2.5. IoT security

Current designing solutions of the Internet of things concept, are based on use a large number of sensors collecting and passing data, measurement devices, computing devices, games consoles, smart phones, appliances with embedded processors running applications, etc. Smart things dial with providing domestic functions, entertainment, health care, appliance and application control, etc. (Moosavi et al., 2016) [59], (Rahmani et al., 2017) [60].

Various studies are performed for identifying potential privacy, security and reliability risks and needs in such highly interconnected environments. For example, CIA director David Petraeus said that the use of data from Internet-connected household appliances can be used to compile a detailed dossier on a person, which will reveal his weaknesses and habits (Sarma et al., 2002) [61].

Given the widespread adoption of Radio-Frequency IDentification (RFID) tag technology, it is potentially possible privacy violations due to remote monitoring of the moving objects (Weis et al., 2004) [62], (Khan et al., 2017) [63].

On the one hand, the use of IoT leads to increased risks of violating the privacy and leakage of confidential commercial data. On the other hand, it leads to increased risks of DDoS attacks. According to the analysis of the DDoS attack occurred on October 21, 2016, it became possible due to the use of a large number of smart home appliances (DDoS 2016) [64].

Security services are fundamental to handling confidentiality, authentication, integrity, authorization, etc., and can be implemented by a combination of cryptographic mechanisms, such as block ciphers, hash functions, or signature algorithms, and non-cryptographic mechanisms.

The main mechanism of ensuring data security in the world of Internet of Things is the so-called LightWeight Cryptography (LWC). Due to things are mostly operated autonomously, to ensure data security, it is required to use a special approach based on LWC. Among the first studies in the field, the works of (Weis et al., 2004) [62], (Sarma et al., 2002) [61] define technical requirements for LWC.

A joint meeting of the Working Group on Intellectual Networking Architecture and Computer Security Working Group of the National Institute of Standards and Technologies, including Smart Grid Interoperability Panel — Cyber Security Working Group (NIST SGIP-CSWG), concerns the need for research in the field of LWC to implement cryptographic protection of millions of devices equipped with limited computing resources and technological restrictions. Due to operating and functioning conditions, as well as the price reduction, these devices have significant limitations on the memory, computing power, power consumptions, etc.

Since proposing the IoT concept in the late 1990s, security experts warn of the potential risk of a large number of unprotected devices connected to the Internet,

In December 2013, Proofpoint, an enterprise security company, discovered the first IoT botnet. According to Proofpoint, more than 25% of the bots are running on devices other than computers, including smart TVs, children's monitors, and other household appliances.

Data is exchanged over the local wireless network or via the Internet. Wireless network allows IoT to manage remote devices (bulbs, a kettle, video camera, etc.) without installing additional communication infrastructure, but it imposes increased requirements for ensuring an appropriate level of security.

For example, strict limitations are imposed on the energy consumption of cryptographic algorithms for passive smart devices, such as RFID tags or contactless smart cards.

In accordance with the ISO/IEC standard (Hodjat & Verbauwhede, 2004) [65], passive RFID tags should have an energy

Please cite this article in press as: N. Chervyakov, et al., AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security, Future Generation Computer Systems (2017), https://doi.org/10.1016/j.future.2017.09.061.

consumption level of not more than 15 μW in order to guarantee operation of the device within a radius of 1 m. However, this restriction limits the performance of devices.

Many of the requirements for algorithms for low-resource environments have been defined within the framework of the international standard ISO/IEC FDIS 29192 – Information technology – Security techniques - Lightweight cryptography.

ISO/IEC 29192 defines the facilities of low-resource cryptography to provide data confidentiality, authentication, identification, non-repudiation, and key exchange.

The main characteristics of the cryptographic algorithms are the complexity and speed. Speed is a very important for many (but not all) applications. It depends not only on the frequency of the processor, but also on the number of cores (in the case of hardware implementation), as cryptographic primitives are usually very suitable for parallelization.

We consider various options for the efficient implementation of Advanced Encryption Standard (AES).

AES hardware implementation. The fastest implementation of the AES algorithm demonstrates a speed of up to 70 Gb/s (Hodjat & Verbauwhede, 2004) [65]. This implementation uses the processor pipeline architecture and requires more than 250,000 GE (Gigabit Ethernet). At the same time, the most compact implementation of this algorithm requires about 2400 GE (Moradi et al., 2011) [66].

AES software implementation. For standard processors, there is an implementation of the AES, which provides a speed of 7.6 clock cycles per byte on an Intel Core 2 Q9550 processor, and 6.9 clock cycles per byte on an Intel Core i7 processor (Käsper & Schwabe 2009) [67].

AES hardware and software implementation. The adoption of the AES standard caused the development of additional commands for Intel family processors. A similar extension of PadLock engine exists in microprocessors from VIA Technologies. The purpose of this extension is to accelerate applications using AES encryption, which provides an encryption rate of about 0.75 clock cycles per byte (Preneel, 2010) [68]. However, the use of AES for protection of data transmission over wireless networks requires modification codes and large computational resources to decrypt data, when data from a large number of Internet things are processed on a fog node.

An alternative way is to use a cryptographic primitive, which is an algorithm for ensuring data security, correctness and reliability of data, and, also, allows the data to be processed in the encrypted form.

Mignotte secret sharing scheme satisfies above requirements. This scheme has similar properties to RRNS and provides a secret sharing scheme, detection, localization, error correction code, and homomorphic cipher at the same time.

In edge and fog computing, the main idea is to perform computation locally, close to the source of data. In general, sending data to the clouds is not recommended. However, when the computational tasks cannot be performed due to the technical limitations of devices, and if the data processing requires more energy than the data transfer over a wireless network, the data is moved to the clouds.

3. The RRNS distributed data processing model

To design a reliable and secure data storage based on multi-clouds technology, we use error correction codes in RRNS. The data storage has the following properties: homomorphic cypher, weighted secret sharing scheme and error correction codes.

Let ith cloud corresponds to RRNS module p_i of the form $2^b - \alpha_i$, where b is module length, and α_i is small integer. The values b and α_i are chosen according to available computational resources of ith cloud, demanded level of security, and the reliability (see, Section 4).

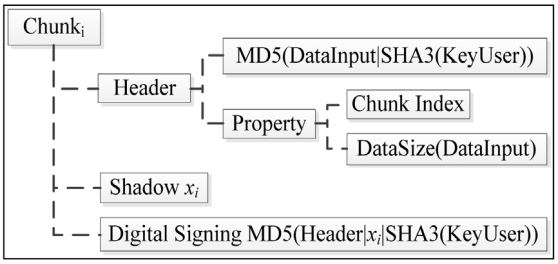

Fig. 1. Chunk description.

The number of clouds is equal to n. For (k, n) settings, we can recover $r = n - k$ data using data from any k clouds. Since the range of RRNS is P, where $L = \lceil \log_2 P \rceil \approx k \cdot b$ denotes the data size.

In this scenario, each cloud provider receives a chunk of data that consists of chunk identifier, chunk properties, projection of the original data, and simplified digital signature. To compute the unique identifier, we use two algorithms: hash function based on MD5 (Wang & Yu, 2005) [69], and on SHA-3 (Pritzker & Gallagher, 2014) [70]. Collisions are avoided and the security is obtained due to SHA-3, while fast image of data is obtained by insecure MD5.

The structure of the chunk is shown in Fig. 1.

4. Configurable parameters

4.1. The probability of information loss

Cloud-based services can crash just like any other type of technology. For example, access to information of Amazon has been limited for a long time due to DDoS attacks in 2009. In 2013, a series of cloud outages was reported for Amazon, Microsoft and Google. Technical failures and data loss due to power outages are reported by Amazon, Dropbox, Microsoft, Google, and Yandex Disk. In the first quarter of 2014, Dropbox experienced service outages twice. Bankruptcy was imposed for cloud storage company Nirvanix in 2013.

To prevent and deal with DDoS attacks, web service Greatfire.org spends up to $30,000 per day (Munson, 2015) [71]. One of the most powerful DDoS attacks occurred in October 2016, according to a report by (Leswing, 2016) [72]. Users could not access their data about 11 h. According to the report of the DDoS attacks in the first quarter of 2016 the longest attack lasted 197 h, or 8.2 days (Kaspersky lab, 2016) [73].

Using the definition of geometric probability, probability of denial access to the service is equal to $8.2/90 \approx 0.09$. Taking the mean between the longest DDoS attack on web service and event when there was no DDoS attacks, we obtain the probability of failure to access data as the result of DDoS attacks equals to 0.05.

Considering reports about technical failures in the cloud services, the probability of loss of information is equal to $3/365 \approx 0.01$ (Gage, 2013) [74], (WCO, 2014) [75], (Wu et al., 2017) [76], etc. Therefore, using the law of addition of probabilities and the Bernoulli formula, we calculate the probability of failure to access the data using a secret sharing scheme (k, n) according to the following formula:

$$Pr\,(k, n) = \sum_{i=n-k+1}^{n} C_n^i \left(0.01^i \cdot 0.99^{n-i} + 0.05^i \cdot 0.95^{n-i}\right).$$

The probability of information loss for various system parameters are shown in Fig. 2.

Please cite this article in press as: N. Chervyakov, et al., AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security, Future Generation Computer Systems (2017), https://doi.org/10.1016/j.future.2017.09.061.

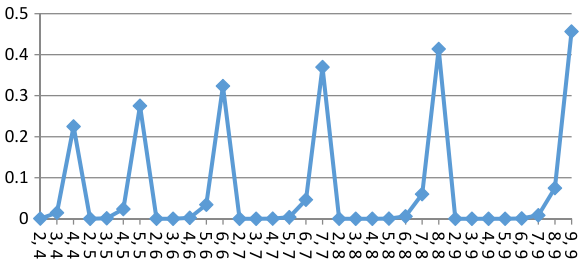

Fig. 2. The probability of information loss versus RRNS settings (k, n).

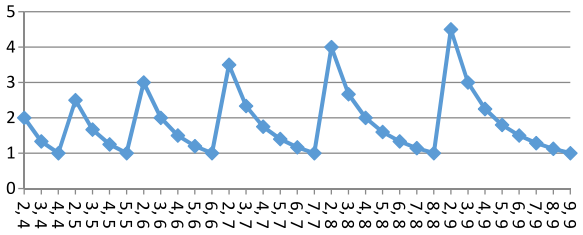

Fig. 3. Data redundancy versus RRNS settings (k, n).

We observe that our data storage scheme provides the least probability of data loss with RRNS settings (k, n) from $(2, n)$ to $(n-2, n)$, where $n = 4, 5, \ldots, 9$. The highest probability of data loss is with RNS settings (n, n).

4.2. Data redundancy

Redundancy in the big data storage is an important issue. Since in the worst case the number of bits that need to be stored $(p_1 - 1, p_2 - 1, \ldots, p_n - 1)$ is approximately equal to $\sum_{i=1}^{n} \log_2 p_i$. The input data is L and approximately equals to $\sum_{i=1}^{k} \log_2 p_i$. We calculate redundancy as the ratio of the stored encoded data and the original data size:

$$\sum_{i=1}^{n} \log_2 p_i / \sum_{i=1}^{k} \log_2 p_i.$$

If RNS moduli satisfy the condition

$$2^{b-1} < p_1 < p_2 < \cdots < p_n < 2^b,$$

then the redundancy satisfies the inequality

$$\frac{\sum_{i=1}^{n}(b-1)}{\sum_{i=1}^{k} b} < \frac{\sum_{i=1}^{n} \log_2 p_i}{\sum_{i=1}^{k} \log_2 p_i} \le \frac{\sum_{i=1}^{n} b}{\sum_{i=1}^{k} b} = \frac{n}{k}.$$

Consequently, the redundancy is roughly n/k. Redundancy versus RRNS parameters is shown in Fig. 3.

Let us consider an example of (4,4) scheme with RRNS moduli set $p_1 = 59, p_2 = 61, p_3 = 63, p_4 = 64$. In this case, the dynamic range is $P = \prod_{i=1}^{4} 14511168$.

Let $X = 14511140$ and it has 24 bits. X has the following representation $X \overset{RRNS}{\rightarrow} (31, 33, 35, 36)$. 31 is a 5 bits number, 33, 35, 36 are 6 bits numbers. Therefore, the redundancy is $(5+3\cdot 6)/24 = 23/24 < 1$.

We see that the redundancy has minimal values with RRNS settings (n, n), where $n = 4, 5, \ldots, 9$, and less than those of the Bigtable system $(\lceil (n+1)/3 \rceil, n)$.

4.3. Speed of data encoding

To analyze the speed of data encoding, we use the minimum technical characteristics of VM provided by Microsoft Azure. It is Intel Xeon® E5-2673 v.3, 2 GB of RAM, 16 GB SSD hard drive. It has an average speed 2^{30} bit operations per second according to tests from (Geekbench Browser site) [77].

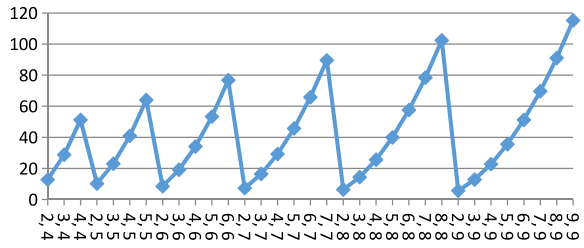

Fig. 4. The speed of data encoding (Mb/s) versus RRNS settings (k, n).

The coding rate changes depending on the processor performance and number of cores. In case of z cores, the speed of algorithms is increased by z times.

The main objective of data encryption algorithms proposed by Chervyakov et al. (2016) [58] is to find the residue of division of L-bit word by RRNS modulus of the form $2^{b_i} \mp \alpha_i$. If $b_1 = b_2 = \cdots = b_n = b$, then, effective algorithm is based on the neural network of the finite ring.

Its latest modification from work Chervyakov et al. (2016) [58] allows to achieve $O(b \cdot \log_2 k)$ algorithmic complexity. Hence, we assume that calculation of the remainder of division requires roughly $b \cdot \log_2 k$ bit operations.

To represent L-bit number in RNS, it is required to perform an operation of finding the remainder of division by RNS modulus n times. Thus, total number of bit operations is $n \cdot b \cdot \log_2 k$. Since the size of the input block is $L \approx k \cdot b$ bits, the number of blocks in 1 Mb is $2^{23}/L \approx 2^{23}/(k \cdot b)$. Therefore, in order to encode 1 Mb of data, $2^{23} \cdot n \cdot b \cdot \log_2 k/(k \cdot b) = 2^{23} \cdot n \cdot \log_2 k/k$ bit operations are required.

The speed of data encoding in MB/s can be calculated by the formula

$$V_C = \frac{2^{30} \cdot k}{2^{23} \cdot n \cdot \log_2 k} = \frac{k \cdot 2^7}{n \cdot \log_2 k}.$$

The dependence of the data encoding speed from the parameters of the scheme is shown in Fig. 4.

We see that the graph is a saw type, where the maximum values of the coding rate are achieved in the schemes (n, n), and minimum coding rates are achieved in the schemes $(2, n)$. The user can select the required parameters to provide the required value of the data coding rate.

4.4. Data decoding rate

When the data is decoded with no errors, according to Chinese Remainder Theorem (CRT) with algorithmic complexity $O(L^2)$, it requires roughly $L^2 \approx k^2 \cdot b^2$ bit operations. In case of r RRNS moduli, we can detect and correct $r - 1$ errors.

To detect and localize the error, we use the algorithm based on projections. To compute a projection, we use CRT, therefore, one projection is computed in roughly $L^2 \approx k^2 \cdot b^2$ bit operations.

Since the number of projections is C_n^{k+1}, to detect and localize an error, we need $C_n^{k+1} \cdot k^2 \cdot b^2$ bit operations. To encode 1 Mb of data, $2^{23} \cdot C_n^{k+1} \cdot k^2 \cdot b^2/(k \cdot b) = 2^{23} \cdot C_n^{k+1} \cdot b \cdot k$ bit operations are required. Therefore, the speed of decoding is

$$V_D = \frac{2^{30}}{2^{23} \cdot C_n^{k+1} \cdot b \cdot k} = \frac{2^7}{C_n^{k+1} \cdot b \cdot k}.$$

The dependence of the data decoding speed from the parameters of the scheme is shown in Fig. 5 for $\{b = 8, 16, 32\}$.

Please cite this article in press as: N. Chervyakov, et al., AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security, Future Generation Computer Systems (2017), https://doi.org/10.1016/j.future.2017.09.061.

5. Approximation of the rank of RNS number

In this section, we propose the method of RRNS to binary conversion based on Approximation of the Rank (AR) of RNS number based on replacement computation.

The suggested approach reduces quantity of calculated projections of the number, and replaces computationally complex operation of division of long integers by taking the least significant bits. It reduces the complexity from $O(L \cdot \log L \cdot \log\log L)$ to $O(1)$.

The rank of the number in RNS is determined according to the CRT, the value X can be calculated by the Eq. (2):

$$X = \sum_{i=1}^{n} P_i \left|P_i^{-1}\right|_{p_i} x_i - r_X \cdot P, \tag{2}$$

where

$$r_X = \left\lfloor \sum_{i=1}^{n} \frac{\left|P_i^{-1}\right|_{p_i}}{p_i} x_i \right\rfloor, P = \prod_{i=1}^{n} p_i, P_i = P/p_i$$

for all $i = \overline{1,n}$, and r_X is a rank of X, which is a positive integer that shows how many times the dynamic range of the RNS can be increased.

From Eq. (2), it follows that to compute r_X, we need to perform expensive operation of integer division or use real numbers with N digits accuracy.

For efficient computing of the rank, we use an approach based on approximate method and modular adder, which decreases the demanded accuracy of computations:

$$R_X = \left\lfloor \sum_{i=1}^{n} k_i x_i / 2^N \right\rfloor, \tag{3}$$

where $k_i = \left\lceil \left|P_i^{-1}\right|_{p_i} 2^N / p_i \right\rceil$.

Now, we derive Theorem 1 that shows how values N, R_X, and r_X are related. It provides a theoretical basis of our approach.

Theorem 1. *If $N = \lceil \log_2 \rho \rceil$, then $r_X = R_X$ or $r_X = R_X - 1$, where $\rho = \sum_{i=1}^{n} p_i - n$.*

Proof. Let $k_i = \left\lceil \left|P_i^{-1}\right|_{p_i} 2^N / p_i \right\rceil$, then k_i can be represented as: $k_i = \left|P_i^{-1}\right|_{p_i} 2^N / p_i + \theta_i$, where $0 \le \theta_i \le (p_i - 1)/p_i$.

Let us compute the value of $\sum_{i=1}^{n} k_i x_i$:

$$\begin{aligned} \sum_{i=1}^{n} k_i x_i &= \sum_{i=1}^{n} \left(\frac{\left|P_i^{-1}\right|_{p_i} 2^N}{p_i} + \theta_i \right) x_i \\ &= \sum_{i=1}^{n} \frac{\left|P_i^{-1}\right|_{p_i} 2^N}{p_i} x_i + \sum_{i=1}^{n} \theta_i \cdot x_i \\ &= 2^N \sum_{i=1}^{n} \frac{\left|P_i^{-1}\right|_{p_i}}{p_i} x_i + \sum_{i=1}^{n} \theta_i \cdot x_i \end{aligned} \tag{4}$$

Computing R_X by substituting Eq. (4) in Eq. (3), we obtain:

$$R_X = \left\lfloor \sum_{i=1}^{n} \frac{\left|P_i^{-1}\right|_{p_i}}{p_i} x_i + \frac{\sum_{i=1}^{n} \theta_i \cdot x_i}{2^N} \right\rfloor \tag{5}$$

From Eq. (5) and Eq. (2), it follows that

$$r_X = R_X - 1, \text{ if } \frac{\sum_{i=1}^{n} \theta_i \cdot x_i}{2^N} < 1.$$

The sufficient condition is: $\sum_{i=1}^{n} \theta_i \cdot x_i < 2^N$. Since $\sum_{i=1}^{n} \theta_i \cdot x_i \le \sum_{i=1}^{n} (p_i - 1) = \rho$, then $N = \lceil \log_2 \rho \rceil$ is sufficient to hold the inequality $\sum_{i=1}^{n} \theta_i \cdot x_i < 2^N$. Theorem is proved. □

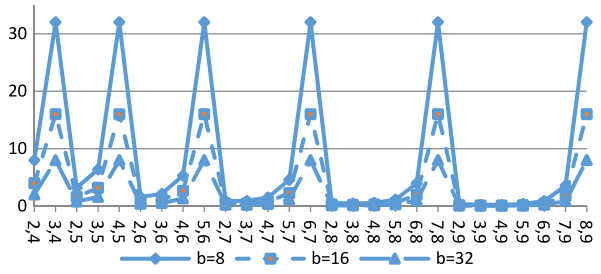

Fig. 5. The speed of data decoding (Mb/s) versus RRNS settings (k, n) for $\{b = 8, 16, 32\}$.

Complexity of the algorithm to calculate the rank of r_X based on Theorem 1 is $O(b \cdot \log b)$. Since coefficients k_i are of size $b + \lceil \log_2 n \rceil$ bits, we are able to compute the value of X efficiently.

The proposed algorithm allows to compute values in the same range $-P < X < P$ as Montgomery modular multiplication algorithm. However, Montgomery modular multiplication is inefficient for the given problem. It requires converting each integer from binary number system to Montgomery system, which has the same complexity as computation of X.

Example 1. Let RNS moduli set be $p_1 = 2, p_2 = 3, p_3 = 5$. Dynamic range of RNS is $P = 2 \cdot 3 \cdot 5 = 30$. We convert $X = 8 \overset{RNS}{\rightarrow} (0, 2, 3)$, $Y = 29 \overset{RNS}{\rightarrow} (1, 2, 4)$ from RNS to binary number system using Theorem 1.

1. RNS constants are computed once and kept in memory.

$P_1 = P/p_1 = 15, P_2 = P/p_2 = 10, P_3 = P/p_3 = 6;$

$\left|P_1^{-1}\right|_{p_1} P_1 = 15, \left|P_2^{-1}\right|_{p_2} P_2 = 10, \left|P_3^{-1}\right|_{p_3} P_3 = 6;$

$\rho = -3 + 2 + 3 + 5 = 7, N = \lceil \log_2 7 \rceil = 3;$

$k_1 = \left\lceil \left|P_1^{-1}\right|_{p_1} 2^N / p_1 \right\rceil = 4, k_2 = \left\lceil \left|P_2^{-1}\right|_{p_2} 2^N / p_2 \right\rceil = 3,$

$k_3 = \left\lceil \left|P_3^{-1}\right|_{p_3} 2^N / p_3 \right\rceil = 2;$

2. To calculate X and Y, we compute the sums:

$\sum_{i=1}^{3} k_i x_i = 4 \cdot 0 + 3 \cdot 2 + 2 \cdot 3 = 12.$

$\sum_{i=1}^{3} k_i y_i = 4 \cdot 1 + 3 \cdot 2 + 2 \cdot 4 = 18.$

Using Eq. (3), we obtain:

$$R_X = \left\lfloor \frac{\sum_{i=1}^{3} k_i x_i}{2^N} \right\rfloor = 1, \quad R_Y = \left\lfloor \frac{\sum_{i=1}^{3} k_i x_i}{2^N} \right\rfloor = 2.$$

Finally, we compute X and Y using Eq. (2), and Theorem 1.

$$\begin{aligned} X &= \sum_{i=1}^{3} P_i \left|P_i^{-1}\right|_{p_i} x_i - R_X P = \\ &= 15 \cdot 0 + 10 \cdot 2 + 6 \cdot 3 - 30 = 8. \end{aligned}$$

$$\begin{aligned} Y^* &= \sum_{i=1}^{3} P_i \left|P_i^{-1}\right|_{p_i} y_i - R_Y P = \\ &= 15 \cdot 1 + 10 \cdot 2 + 6 \cdot 4 - 2 \cdot 30 = -1 \end{aligned}$$

When $Y^* < 0$, then $Y = Y^* + P = -1 + 30 = 29$, otherwise, $Y = Y^*$.

As shown in Example 1, the rank of X is equal to the true value of the function, but approximate rank of Y is less than the true value of the rank.

Computation of $\sum_{i=1}^{3} P_i \left|P_i^{-1}\right|_{p_i} x_i$ can be done in parallel with computation of AR. Since its computational complexity is equal to the complexity of $R_X \cdot P$, they take approximately the same time to be performed.

Please cite this article in press as: N. Chervyakov, et al., AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security, Future Generation Computer Systems (2017), https://doi.org/10.1016/j.future.2017.09.061.

Therefore, correct implementation of decoding algorithm allows to increase the speed of the algorithm. If the approximate rank R_X of X in RRNS equals to r_X+1 (the true rank plus 1), then the value of X is less than zero. In that case, we should add the dynamic range of the system to the negative value of X.

6. Error detection, correction and control of arithmetic operations

6.1. Method of error detection, localization and correction

Using Theorem 1, we propose a new method for data decoding based on Approximation of the Rank and Error Correcting Codes (AR-ECC).

From Eq. (2), it follows that $\sum_{i=1}^{n} k_i x_i$ can be represented in the form:

$$\sum_{i=1}^{n} k_i x_i = X + r_X P \tag{6}$$

Assume that an error $E \overset{RRNS}{\rightarrow} (e_1, e_2, \ldots, e_n)$ occurred during computations, and the user (storage or communication) obtained the value $X + E$ instead of X. Then by Eq. (6), we have:

$$\sum_{i=1}^{n} k_i (x_i + e_i) = X + E + r_X P + r_E P$$

Without loss of generality, we assume that RRNS moduli set are in ascending order i.e. $p_1 < p_2 < \cdots < p_n$. Since in the proposed RRNS, k moduli are included in the dynamic range, and r is redundant (control moduli), where $k + r = n$, then

$$X < \prod_{i=1}^{k} p_i = R.$$

The value of $\lfloor E/R \rfloor$ is $\lfloor (X+E)/R \rfloor$ or $\lfloor (X+E)/R \rfloor - 1$. If $\lfloor (E+X)/R \rfloor = 0$, then $E = 0$.

Therefore, we can use the value $\lfloor (E+X)/R \rfloor$ to determine if the result is correct, i.e. if there is or there is no error.

Due to the fact that an error is of the form: $E = \beta P_I$, where $P_I = \prod_{i \in I} p_i$, β is integer in the interval $[0, P/P_I - 1]$, and I is set of RRNS moduli that do not have an error. The value $\lfloor E/R \rfloor$ can be used as an error syndrome, where each $\lfloor E/R \rfloor$ is unambiguously defined by E and I.

We do some precomputations: sort the values of all possible errors $\lfloor E/R \rfloor$ in ascending order and map to E. If we use binary search in a sorted array of values $\lfloor E/R \rfloor$, we find E and set I in logarithmic of array size time.

Let $X' = X + E$ and $X' \overset{RRNS}{\rightarrow} (x'_1, x'_2, \ldots, x'_n)$. Using Eq. (2), we compute

$$\left\lfloor \frac{X'}{R} \right\rfloor = \left\lfloor \frac{\sum_{i=1}^{n} \left| P_i^{-1} \right|_{p_i} \cdot P_i \cdot x'_i - r_{X'} P}{R} \right\rfloor \tag{7}$$

Since P/R is integer then according to the property of floor, integer can be taken out as a common factor, and the Eq. (7) can be rewritten:

$$\left\lfloor \frac{X'}{R} \right\rfloor = \left\lfloor \frac{\sum_{i=1}^{n} \left| P_i^{-1} \right|_{p_i} \cdot P_i \cdot x'_i}{R} \right\rfloor - \frac{r_{X'} P}{R} \tag{8}$$

Let $P/R = M$. Due to the fact that $0 \le \lfloor X'/R \rfloor < M$, Eq. (8) is equivalent to

$$\begin{aligned} \left\lfloor \frac{X'}{R} \right\rfloor &= \left| \left\lfloor \frac{\sum_{i=1}^{n} \left| P_i^{-1} \right|_{p_i} \cdot P_i \cdot x'_i}{R} \right\rfloor - \frac{r_{X'} P}{R} \right|_M \\ &= \left| \left\lfloor \frac{\sum_{i=1}^{n} \left| P_i^{-1} \right|_{p_i} \cdot P_i \cdot x'_i}{R} \right\rfloor \right|_M \end{aligned} \tag{9}$$

From the definition of P_i, it follows that for all $i = \overline{k+1, n}$, the value P_i/R is integer. Then, according to the property of floor, integer can be taken out as a common factor, and the Eq. (9) can be rewritten:

$$\left\lfloor \frac{X'}{R} \right\rfloor = \left| \left\lfloor \frac{\sum_{i=1}^{k} \left| P_i^{-1} \right|_{p_i} \cdot P_i \cdot x'_i}{R} \right\rfloor + \sum_{i=k+1}^{n} \frac{\left| P_i^{-1} \right|_{p_i} \cdot P_i}{R} \cdot x'_i \right|_M \tag{10}$$

In Eq. (10), the value of $\left\lfloor \sum_{i=1}^{k} \left| P_i^{-1} \right|_{p_i} \cdot P_i \cdot x'_i / R \right\rfloor$ can be computed according to Theorem 1. For all $i = \overline{k+1, n}$, the value $\left| P_i^{-1} \right|_{p_i} \cdot P_i / R$ are precomputed constants.

Example 2. Let scheme be (3, 5) and moduli set be $p_1 = 2, p_2 = 3, p_3 = 5, p_4 = 7, p_5 = 11$. Parameters of RRNS are $R = 2 \cdot 3 \cdot 5$ and $P = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 = 2310$. There is a number represented in RRNS $X = 8 \overset{RRNS}{\rightarrow} (0, 2, 3, 1, 8)$.

We consider how to use the proposed approach to detect, localize and correct an error.

Precomputations: $M = P/R = 77$, $P_1 = P/p_1 = 1155$, $P_2 = P/p_2 = 770$, $P_3 = P/p_3 = 462$, $P_4 = P/p_4 = 330$, $P_5 = P/p_5 = 210$, $\left| P_1^{-1} \right|_{p_1} P_1 = 1155$, $\left| P_2^{-1} \right|_{p_2} P_2 = 2 \cdot 770 = 1540$, $\left| P_3^{-1} \right|_{p_3} P_3 = 3 \cdot 462 = 1386$, $\left| P_4^{-1} \right|_{p_4} P_4 = 1 \cdot 330 = 330$, $\left| P_5^{-1} \right|_{p_5} P_5 = 1 \cdot 210 = 210$.

We make a table of possible errors depending on the different values of $\lfloor W/R \rfloor$:

- 0: there is no error;
- 38, 39: error is in w_1;
- 25, 26, 51, 52: error is in w_2;
- 15, 16, 30, 31, 46, 47, 61, 62: error is in w_3;
- 11, 22, 33, 44, 55, 66: error is in w_4;
- 7, 14, 21, 28, 35, 42, 49, 56, 63: error is in w_5.

Let the error vector E equals to $E = (0, 0, 0, 1, 0)$, then we obtain $X' = X + E \overset{RRNS}{\rightarrow} (0, 2, 3, 2, 8)$, we compute $\lfloor X'/R \rfloor$ using Eq. (10), and we get:

$$\begin{aligned} \left\lfloor \frac{X'}{R} \right\rfloor &= \left| \left\lfloor \frac{1155 \cdot 0 + 1540 \cdot 2 + 1386 \cdot 3}{30} \right\rfloor + 11 \cdot 2 + 7 \cdot 8 \right|_{77} \\ &= 11. \end{aligned}$$

Since $\lfloor X'/R \rfloor = 11$, the error is in w_4, and $E = (0, 0, 0, 1, 0)$. Therefore, the value of X is $X \overset{RRNS}{\rightarrow} (0, 2, 3, 1, 8)$.

We apply Theorem 1 to compute $\left\lfloor \sum_{i=1}^{k} \left| P_i^{-1} \right|_{p_i} \cdot P_i \cdot x'_i / R \right\rfloor$.

$k_1 = \left\lceil 2^3 \left| P_1^{-1} \right|_{p_1} \cdot P_1 / 30 \right\rceil = 308,$

$k_2 = \left\lceil 2^3 \left| P_2^{-1} \right|_{p_2} \cdot P_2 / 30 \right\rceil = 411,$

$k_3 = \left\lceil 2^3 \left| P_3^{-1} \right|_{p_3} \cdot P_3 / 30 \right\rceil = 370.$

$\left\lfloor \sum_{i=1}^{k} \left| P_i^{-1} \right|_{p_i} \cdot P_i \cdot x'_i / R \right\rfloor = \left\lfloor \sum_{i=1}^{3} k_i x'_i / 8 \right\rfloor = \lfloor (308 \cdot 0 + 411 \cdot 2 + 370 \cdot 3)/8 \rfloor = 241.$

6.2. Error control of RRNS arithmetic operations

In this section, we introduce AR-RRNS method for error detection, correction, and controlling computational results.

The basic arithmetic operations in most of the algorithms are addition, subtraction, multiplication and division. The right shift is a division operation into a power of two and the left shift is an operation of multiplying by a power of two. For instance, unary

algebraic operation of negation "$-a$" can be represented as binary algebraic operation using the expression $0 - a$.

If we denote ith binary operation O_i, with operands X_i and Y_i, it can be written as: $O_i = f_i(X_i, Y_i)$.

We denote the sequence of binary algebraic operations O_i performed consequently over operands $X_1, X_2, \ldots, X_m$ as $F(X_1, X_2, \ldots, X_m)$.

Since the control of results of the intermediate computations is a complex problem, and probability of errors in computations is high, we need a method to check the result of $F(X_1, X_2, \ldots, X_m)$.

To control results of computations, AN codes can be used (Grangetto et al., 2005) [78]. However, they require redundant information, and do not adapted to distributed systems. RRNS allows to control computations with less redundant information.

Let the true result of $F(X_1, X_2, \ldots, X_m)$ be in the form: $T \overset{RRNS}{\rightarrow} (t_1, t_2, \ldots, t_n)$. Suppose, obtained result is $A \overset{RRNS}{\rightarrow} (a_1, a_2, \ldots, a_n)$. If for all $i = \overline{1, n}$, $t_i = a_i$ holds, then the obtained result is correct, otherwise, is wrong. To check if the result of $F(X_1, X_2, \ldots, X_m)$ is correct, we use the method of scaling and base extension in RRNS.

For instance, from first $n-1$ residues $(a_1, a_2, \ldots, a_{n-1})$, we compute $\overline{a_n}$. If $\overline{a_n} = a_n$, then the result of $F(X_1, X_2, \ldots, X_m)$ is correct, otherwise, is wrong.

To find the true result, we use the method of error detection and localization considered above. Hence, the results have no errors that cannot be detected in RRNS.

The error can be corrected not only in the case of overlapping moduli errors during computations over one modulus, but also in case of a single failure.

For example, if an error occurred in p_i, but, during the computation of $F(X_1, X_2, \ldots, X_m)$, there is multiplication by zero the error is corrected.

Theorem 2. *Let $p_1, p_2, \ldots, p_n$ be RRNS moduli set, $X \overset{RRNS}{\rightarrow} (x_1, x_2, \ldots, x_n)$ and $Y \overset{RRNS}{\rightarrow} (y_1, y_2, \ldots, y_n)$ be numbers in RRNS with ranks r_X and r_Y, then the following equality holds:*

$$r_{X+Y} = r_X + r_Y - \sum_{x_i+y_i \geq p_i} \left|P_i^{-1}\right|_{p_i} \tag{11}$$

Proof. By definition of rank r_{X+Y} is

$$r_{X+Y} = \left\lfloor \sum_{i=1}^{n} \frac{\left|P_i^{-1}\right|_{p_i} |x_i + y_i|_{p_i}}{p_i} \right\rfloor \tag{12}$$

Considering that for all $\forall i \in \overline{1, n}$ the following equality holds

$$|x_i + y_i|_{p_i} = \begin{cases} x_i + y_i - p_i, & if x_i + y_i \geq p_i, \\ x_i + y_i, \text{ other} \end{cases}$$

then Eq. (12) can be rewritten in the form:

$$r_{X+Y} = \left\lfloor \sum_{i=1}^{n} \frac{\left|P_i^{-1}\right|_{p_i} x_i}{p_i} + \sum_{i=1}^{n} \frac{\left|P_i^{-1}\right|_{p_i} y_i}{p_i} - \sum_{x_i+y_i \geq p_i} \left|P_i^{-1}\right|_{p_i} \right\rfloor \tag{13}$$

Since integer A can be represented as sum of whole and fractional part, i.e. $A = \lfloor A \rfloor + \{A\}$, then, according to the property of floor, integer part is a common factor. Consequently, Eq. (13) can be rewritten in the following form:

$$\begin{aligned} r_{X+Y} = &\left\lfloor \sum_{i=1}^{n} \frac{\left|P_i^{-1}\right|_{p_i} x_i}{p_i} \right\rfloor + \left\lfloor \sum_{i=1}^{n} \frac{\left|P_i^{-1}\right|_{p_i} y_i}{p_i} \right\rfloor - \sum_{x_i+y_i \geq p_i} \left|P_i^{-1}\right|_{p_i} \\ &+ \left\lfloor \left\{ \sum_{i=1}^{n} \frac{\left|P_i^{-1}\right|_{p_i} x_i}{p_i} \right\} + \left\{ \sum_{i=1}^{n} \frac{\left|P_i^{-1}\right|_{p_i} y_i}{p_i} \right\} \right\rfloor \end{aligned} \tag{14}$$

According to Chervyakov et al. (2014) [79], $\left\{ \sum_{i=1}^{n} \left|P_i^{-1}\right|_{p_i} x_i/p_i \right\} = X/P$ and $\left\{ \sum_{i=1}^{n} \left|P_i^{-1}\right|_{p_i} y_i/p_i \right\} = Y/P$.

If we substitute the values $r_X = \left\lfloor \sum_{i=1}^{n} \left|P_i^{-1}\right|_{p_i} x_i/p_i \right\rfloor$ and $r_Y = \left\lfloor \sum_{i=1}^{n} \left|P_i^{-1}\right|_{p_i} y_i/p_i \right\rfloor$ in Eq. (14), we obtain:

$$r_{X+Y} = r_X + r_Y - \sum_{x_i+y_i \geq p_i} \left|P_i^{-1}\right|_{p_i} + \left\lfloor \frac{X}{P} + \frac{Y}{P} \right\rfloor \tag{15}$$

Since the condition of RRNS applicability is $X + Y < P$, then $\lfloor X/P + Y/P \rfloor = 0$, and Eq. (15) equals to Eq. (11).

Theorem is proved. □

Example 3. Let RNS moduli set be $p_1 = 2, p_2 = 3, p_3 = 5$, with dynamic range of RNS $P = 2 \cdot 3 \cdot 5 = 30$. Two numbers in RNS are $X \overset{RNS}{\rightarrow} (0, 2, 3)$ and $Y \overset{RNS}{\rightarrow} (1, 2, 4)$. We consider several cases how to use Theorem 2 to verify the result. We use coefficients for the given RNS that are computed in Example 1: $r_X = 1, r_Y = 1, \left|P_1^{-1}\right|_{p_1} = 1, \left|P_2^{-1}\right|_{p_2} = 1, \left|P_3^{-1}\right|_{p_3} = 1, \left|P_1^{-1}\right|_{p_1} P_1 = 15, \left|P_2^{-1}\right|_{p_2} P_2 = 10, \left|P_3^{-1}\right|_{p_3} P_3 = 6$.

Case 1. $W = X + Y \overset{RNS}{\rightarrow} (1, 1, 2)$. We compute the sum $\sum_{i=1}^{3} k_i w_i = 4 \cdot 1 + 3 \cdot 1 + 2 \cdot 2 = 11$, then $R_W = \lfloor 11/8 \rfloor = 1$. $W = \sum_{i=1}^{3} P_i \left|P_i^{-1}\right|_{p_i} w_i - R_W P = 15 \cdot 1 + 10 \cdot 1 + 6 \cdot 2 - 30 = 7$. Therefore, according to Theorem 1, we get: $r_W^* = R_W = 1$. We check if the result is correct with Theorem 2 and Eq. (11). We get: $r_W = r_X + r_Y - \left|P_2^{-1}\right|_{p_2} - \left|P_3^{-1}\right|_{p_3} = 1 + 1 - 1 - 1 = 0$.

Since $r_W^* \neq r_W$, there is an error. In this case, $X = 8$ and $Y = 29$, and $W = X + Y = 8 + 29 = 37$. Therefore, there is RNS dynamic range overflow, which lead to the error detection.

Case 2. $W = X + X + E \overset{RNS}{\rightarrow} (1, 1, 1)$, where E is the error vector equals to $E \overset{RNS}{\rightarrow} (1, 0, 0)$. We compute the sum $\sum_{i=1}^{3} k_i w_i = 4 \cdot 1 + 3 \cdot 1 + 2 \cdot 1 = 9$, then $R_W = \lfloor 9/8 \rfloor = 1$. $W = \sum_{i=1}^{3} P_i \left|P_i^{-1}\right|_{p_i} w_i - R_W P = 15 \cdot 1 + 10 \cdot 1 + 6 \cdot 1 - 30 = 1$.

Therefore, according to Theorem 1, we get $r_W^* = R_W = 1$. We check if the result is correct with Theorem 2 and Eq. (11). We get: $r_W = r_X + r_Y - \left|P_2^{-1}\right|_{p_2} - \left|P_3^{-1}\right|_{p_3} = 1 + 1 - 1 - 1 = 0$. Since $r_W^* \neq r_W$, there is an error.

Case 3. $W = X + X \overset{RNS}{\rightarrow} (0, 1, 1)$. We compute the sum $\sum_{i=1}^{3} k_i w_i = 4 \cdot 0 + 3 \cdot 1 + 2 \cdot 1 = 7$, then $R_W = \lfloor 7/8 \rfloor = 0$. $W = \sum_{i=1}^{3} P_i \left|P_i^{-1}\right|_{p_i} w_i - R_W P = 15 \cdot 0 + 10 \cdot 1 + 6 \cdot 1 = 16$. According to Theorem 1, we get: $r_W^* = R_W = 0$. We check if the result is correct with Theorem 2 and Eq. (11). We get: $r_W = r_X + r_Y - \left|P_2^{-1}\right|_{p_2} - \left|P_3^{-1}\right|_{p_3} = 1 + 1 - 1 - 1 = 0$. Since $r_W^* = r_W$, the result is correct.

As it is shown in Example 3, Theorem 2 can be used to verify the result of arithmetic operations even with the minimal redundancy (n, n). However, in the case of an error, it cannot be corrected. In Example 2, we consider the approach that not only detects an error, but also corrects it.

7. Configurable model

Methods for detecting failures in distributed data storage and communication media are typically based on error correction codes, erasure codes, regeneration codes and their modifications. However, if they do not allow to control computations, and do not have the property of homomorphism of arithmetic operations, they require extreme computational power for data processing and analysis.

Please cite this article in press as: N. Chervyakov, et al., AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security, Future Generation Computer Systems (2017), https://doi.org/10.1016/j.future.2017.09.061.

In contrast to the existing methods, error correction codes in RRNS can effectively detect, correct errors, and control computations. They are fully homomorphic that makes them applicable for big data storage and processing.

However, the error correction codes in RRNS have one significant drawback: the high complexity of detection and localization of errors.

To solve this problem, we propose the approximate method to compute ranks of numbers, sort the array of relative values, and use the binary search.

Our AR-ECC approach reduces complexity from linear to the logarithmic of the power of the set of all possible numbers projections in RRNS.

The model is configurable. To determine and configure parameters, the following optimization criteria should be taken into account: accuracy, scalability, reliability, confidentiality, security, and performance.

Accuracy. The accuracy of computations depends on the parameter N in the number rank computation. The precision $N \geq \log_2(\rho \cdot P)$ allows obtaining the rank of a number accurately (Chervyakov et al., 2014) [79]. However, according to Theorem 1, the precision $N = \lceil \log_2(\rho) \rceil$ allows to effectively approximate the rank of a number with a deviation of less than 1 from the true result.

Scalability and reliability. If an error occurs, we use the cloud scalability property to maintain certain level of reliability. We restore or correct the lost data chunks stored in one or several clouds using AR-ECC. We determine the parameters of storage system for desirable reliability using the approach described in Section 4.1, and the amount of memory necessary to store data using the redundancy coefficient from Section 4.2.

Confidentiality and security. To make the scheme asymptotically ideal, Barzu et al. (2013) [80] proposed to use the Asmuth–Bloom algorithm that provides a high degree of data security. However, this algorithm is inapplicable for distributed storage, since it requires redundancy of data storage as in the Shamir scheme.

In RRNS, the computational security of the system depends on the parameters k, n, b_i and α_i. It can be estimated using the formula $V/V_{\bar{I}}$, where V is the volume of data and $V_{\bar{I}}$ is the maximal amount of data that can be leaked to an unauthorized user.

For example, if for all $i = \overline{1, n}$, the condition $b_i = const$ holds, and there is a conspiracy of $k - 1$ clouds, then the cardinality of the set of all possible combinations of input data is greater than or equal to 2^{b-1}.

To estimate and adapt the security level, we use the approach from Section 4.1. It estimates the risks of cloud conspiracy or DDoS attacks on cloud providers according to the chosen level of security.

Performance. An important issue is to minimize the computational costs associated with the implementation of arithmetic operations. Since they are modular, to select RRNS moduli, we have to take into account that moduli are pairwise co-prime integers. Modulo is $2^{b_i} \pm \alpha_i$, where b_i corresponds to the amount of stored and processed data in the i-th cloud, α_i is an integer in binary representation with a fixed number of bits equal to "1".

The approach from Section 4.3 estimates the speed of data encoding and decoding. If there are no errors in the data storage system, then the speed of decoding increases significantly, because there is no need to perform the expensive operation of finding the error.

Section 6.1 describes how to increase the speed of data decoding due to the optimization of the method of finding the error using AR. To compute Eq. (10), we use CRT. It takes, roughly, $M^2 \approx r^2 \cdot b^2$ bit operations, where $r = n - k$. Since the value of Eq. (10) is M, to detect and localize an error, we need $\log_2 M \cdot r \cdot b$ bit operations. In order to encode 1 Mb of data, $2^{23}\left(\log_2 M \cdot r \cdot b + r^2 \cdot b^2\right)/(k \cdot b) = 2^{23} \cdot (\log_2 (r \cdot b) \cdot r + r^2 \cdot b)/k$ bit operations are required. Therefore, the speed of decoding is

$$V_D = \frac{2^{30} \cdot k}{2^{23} \cdot \left(\log_2 (r \cdot b) \cdot r + r^2 \cdot b\right)} = \frac{2^7 \cdot k}{\log_2 (r \cdot b) \cdot r + r^2 \cdot b}.$$

This method increases the speed of data decoding from

$$V_D = \frac{2^7}{C_n^{k+1} \cdot b \cdot k} \text{ up to } V_D = \frac{2^7 \cdot k}{\log_2 (r \cdot b) \cdot r + r^2 \cdot b}$$

To detect and correct at least one error, k has to satisfy the inequality $k \leq n-1$. Fig. 6 shows the data decoding rate depending on parameters that satisfy this condition.

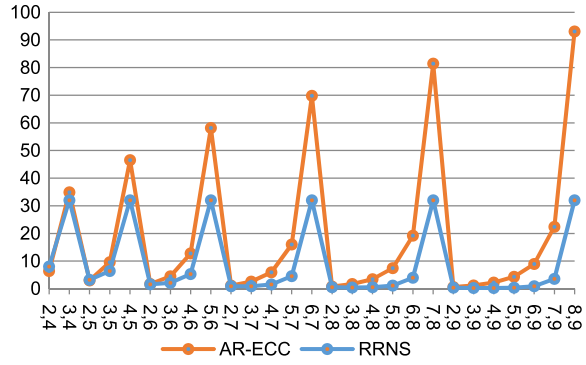

Fig. 6. The speed of decoding data (Mb/s) based on AR-ECC and RRNS on the worst scenario of data recovery with maximum number of errors versus RRNS settings (k, n) for $b = 8$.

To increase the speed of data encoding and decoding, we can use multi-core processors or several VMs. If we choose the values b_i to be multiples of the machine word, the effective implementation of modular arithmetic with distributed operations and neural network of finite ring can be used. It decreases the complexity of finding the remainder of the division from quadratic to linear.

This approach is feasible for modern IoT mobile systems. According to (Geekbench Browser site) [77] performance of single core of Apple A10 Fusion mobile processors is better than performance of the Intel Xeon E5–2673 core.'1.

8. Conclusions

In this paper, we introduce a new configurable data storage scheme based on the error correction codes and secret sharing schemes.

We provide a theoretical basis to calculate probability of information loss, data redundancy, speed of encoding/decoding, and configure parameters to cope with different objective preferences, workloads, and cloud properties.

We show how the proposed scheme allows to configure safety, reliability, and reduce an overhead of data storage by appropriate selection of RRNS parameters.

Based on the proposed approximation rank, we design new method of data decoding AR-ECC that reduces complexity from $O(L^2)$ down to $O(L \cdot \log L)$, and the size of the coefficients from $\lceil \log(\rho \cdot P) \rceil$ to $\lceil \log(\rho) \rceil$ bits.

Using properties of the approximate value and arithmetic properties of RNS, we introduce AR-RRNS method for error detection, correction, and controlling computational results.

However, further study is required to assess its actual efficiency and effectiveness in real systems. This will be the subject of future

Please cite this article in press as: N. Chervyakov, et al., AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security, Future Generation Computer Systems (2017), https://doi.org/10.1016/j.future.2017.09.061.

work providing a comprehensive experimental study of multi-objective optimization with real cloud providers.

Acknowledgments

The work is partially supported by CONACYT (Consejo Nacional de Ciencia y Tecnología, México), grant no. 178415. Part of the work was supported State task No. 2.6035.2017 and Russian Federation President Grant SP-1215.2016.5.

Appendix. Terminology

Reliability is promoted through the use of fault-tolerant systems to be operational in case of components failures. Usually, solutions are based either on full content replication (every file is replicated in every server node) or full distribution (every file is stored in one and only one server node). With full replication, the system is highly reliable and request distribution is easy to implement, as any server node may serve each request. On the other hand, storage scalability is minimal, as the full storage capacity is limited by the server node with the lowest capacity. Furthermore, adding new server nodes to the system does not increase storage capacity.

Scalability is a desirable attribute of a network, system, or process. The concept connotes the ability of a system to accommodate an increasing number of elements or objects, to process growing volumes of work gracefully, and/or to be susceptible to enlargement. When procuring or designing a system, we often require that it be scalable. The requirement may even be mentioned in a contract with a vendor.

Accountability as an essential information security concept means that every user who works with an information system should have specific responsibilities for information assurance. The user tasks are the part of the overall information security plan. They are readily measurable by a person who has managerial responsibility for information assurance.

Availability refers to ensuring that authorized parties are able to access the information when needed. Information has value if the right people can access it at the right times. Denying access to information has become a very common attack nowadays. Almost every week you can find news about high profile websites being taken down by DDoS attacks. The primary aim of DDoS attacks is to deny users of the website access to the resources. Such downtime can be very costly. Other factors that could lead to lack of availability to important information may include accidents such as power outages or natural disasters such as floods. Backup is a key to ensure data availability. Regularly off-site backups can limit the harm caused by damage hard drives or natural disasters. Having an off-site location ready to restore services in case anything happens to the primary data center will heavily reduce the downtime. For information services that are highly critical, redundancy might be appropriate.

Confidentiality is protecting the information from disclosure to unauthorized parties. Information has value, especially in today's world. Bank account statements, personal information, credit card numbers, trade secrets, government documents, etc. Protecting such an information is a very important part of information security

Integrity is protecting information from being modified by unauthorized parties. An information has a value only if it is correct.

Privacy is an approach used for preserving personal data.

References

[1] J. Gantz, D. Reinsel, Extracting value from Chaos state of the universe: An executive summary. IDC iView, 2011. https://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf (Accessed 21 June 2017).
[2] E. Ahmed, M.H. Rehmani, P. Bonnet, Editorial to a special section on information fusion in internet of things, Inform. Sci. 69 (2017) 194.
[3] 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 Circuits Syst. Mag. 15 (2015) 26–44.
[4] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The Hadoop distributed file system, in: 26th IEEE S. Mass Stor. Sys. MSST2010, 2010, pp. 1-10.
[5] S. Ghemawat, H. Gobioff, S.-T. Leung, The google file system, ACM SIGOPS Oper. Syst. Rev. (2003) 29–43.
[6] X. Zhang, C. Liu, S. Nepal, S. Pandey, J. Chen, A privacy leakage upper bound constraint-based approach for cost-effective privacy preserving of intermediate data sets in cloud, IEEE Trans. Parallel Distrib. 24 (2013) 1192–1202.
[7] A.C. Mora, Y. Chen, A. Fuchs, A. Lane, R. Lu, P. Manadhata, Top ten big data security and privacy challenges. Cloud Security Alliance, 2012. https://www.isaca.org/Groups/Professional-English/big-data/GroupDocuments/Big_Data_Top_Ten_v1.pdf (Accessed 21 June 2017).
[8] D. Hubbard, M. Sutton, Top threats to cloud computing v1. 0. Cloud Security Alliance, 2010. https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf. (Accessed 21 June 2017).
[9] S. Singh, P.K. Sharma, S.Y. Moon, J.H. Park, dvanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions, J. Amb. Intel. Hum. Comp. (2017) 1–18.
[10] J.H. Cheon, J. Kim, M.S. Lee, A. Yun, CRT-based fully homomorphic encryption over the integers, Inform. Sci. 310 (2015) 149–162.
[11] A. Tchernykh, U. Schwiegelsohn, E. Talbi, M. Babenko, Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability, J Comput. Sci.-Neth. (2016). http://www.sciencedirect.com/science/article/pii/S1877750316303878.
[12] C. Lynch, Big data: How do your data grow? Nature 455 (2008) 28–29.
[13] A. Fernández, S. del Río, V. López, A. Bawakid, M.J. del Jesus, J.M. Benítez, F. Herrera, Big data with cloud computing: An insight on the computing environment, MapReduce, and programming frameworks, WIRES Data Min. Knowl. 4 (5) (2014) 380–409.
[14] X. Chen, Q. Huang, The data protection of MapReduce using homomorphic encryption, in: Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, 2013, pp. 419–421.
[15] 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.
[16] M.A. Vouk, Cloud computing - issues, research and implementations, CIT, J. Comput. Inf. Technol. 16 (4) (2008) 235–246.
[17] E. Ahmed, M.H. Rehmani, Introduction to the special section on social collaborative internet of things, Comput. Electr. Eng. 58 (2017) 382–384.
[18] F. Chang, J. Dean, S. Ghemawat, W.C. Hsieh, D.A. Wallach, M. Burrows, R.E. Gruber, Bigtable: A distributed storage system for structured data, ACM Trans. Comput. Syst. 26 (2) (2008) 4.
[19] J. Dean, S. Ghemawat, MapReduce: Simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 1–13.
[20] H. Herodotou, H. Lim, G. Luo, N. Borisov, L. Dong, Starfish: A self-tuning system for big data analytics, in: Proceedings of the Fifth Biennial Conference on Innovative Data Systems Research, CIDR, 2011, pp. 261–272.
[21] N. Leavitt, Will NoSQL databases live up to their promise? Computer 43 (2) (2010) 12–14.
[22] L. Okman, N. Gal-Oz, Y. Gonen, E. Gudes, J. Abramov, Security issues in NoSQL databases, in: Proc. 10th IEEE Int. Conf. on Trust, Security and Privacy in Computing and Communications, TrustCom 2011, 8th IEEE Int. Conf. on Embedded Software and Systems, ICESS 2011, 6th Int. Conf. on FCST 2011, 2011, pp. 541–547.
[23] R.L. Rivest, L. Adleman, M.L. Dertouzos, On data banks and privacy homomorphisms, Found. Secure Comput. 4 (11) (1978) 169–180.
[24] C. Gentry, A Fully Homomorphic Encryption Scheme (Ph.D. thesis), Stanford University, 2009.
[25] S. Venugopal, R. Buyya, K. Ramamohanarao, A taxonomy of data grids for distributed data sharing, management, and processing, ACM Comput. Surv. 38 (1) (2006) 3.
[26] M.T. Ozsu, P. Valduriez, Distributed database systems: where are we now? Computer 24 (8) (1991) 68–78.
[27] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, B. Weihl, Globally distributed content delivery, IEEE Internet Comput. 6 (5) (2002) 50–58.
[28] A. Oram, Peer-To-Peer: Harnessing the Power of Disruptive Technologies, first ed., 2001.
[29] A. Ahmad, M.H. Rehmani, H. Tembine, O.A. Mohammed, A. Jamalipour, IEEE access special section editorial: Optimization for emerging wireless networks:

IoT, 5G, and smart grid communication networks, IEEE Access 5 (2017) 2096–2100.
[30] H. Abu-Libdeh, L. Princehouse, H. Weatherspoon, RACS: a case for cloud storage diversity, in: Proceedings of the 1st ACM Symposium on Cloud Computing, SoCC '10, ACM, New York, NY, USA, 2010, pp. 229–240.
[31] A. Adya, W.J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J.R. Douceur, R.P. Wattenhofer, FARSITE: Federated, available, and reliable storage for an incompletely trusted environment, in: Proceedings of the 5th Symposium on Operating Systems Design and Implementation, OSDI, 2002, pp. 1–14.
[32] G. Ateniese, K. Fu, M. Green, S. Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage, ACM Trans. Inform. Syst. Se. 9 (1) (2006) 1–30.
[33] A. Bessani, M. Correia, B. Quaresma, F. André, P. Sousa, DepSky: dependable and secure storage in a cloud-of-clouds, ACM Trans. Storage 9 (4) (2013) 12.
[34] K. Bowers, A. Juels, Oprea, HAIL: A high-availability and integrity layer for cloud storage, in: Proceedings of the 16th ACM Conference on Computer and Communications Security –CCS, 2009, pp. 187-198.
[35] A.G. Dimakis, P.B. Godfrey, Y. Wu, M.J. Wainwright, K. Ramchandran, Network coding for distributed storage systems, IEEE Trans. Inform. Theory 56 (9) (2010) 4539–4551.
[36] Z. Erkin, T. Veugen, T. Toft, R.L. Lagendijk, Generating private recommendations efficiently using homomorphic encryption and data packing, IEEE Trans. Inf. Foren. Sec. 7 (3) (2012) 1053–1066.
[37] C. Gentry, Computing arbitrary functions of encrypted data, Commun. ACM 53 (3) (2010) 97.
[38] M. Gomathisankaran, A. Tyagi, K. Namuduri, HORNS: A homomorphic encryption scheme for cloud computing using residue number system, in: Information Sciences and Systems, CISS, 2011 45th Annual Conference on, 2011, pp. 1–5.
[39] Z. Kong, S.A. Aly, E. Soljanin, Decentralized coding algorithms for distributed storage in wireless sensor networks, IEEE J. Sel. Areas Comm. 28 (2010) 261–267.
[40] M. Li, W. Lou, K. Ren, Data security and privacy in wireless body area networks, IEEE Wirel. Comm. 17 (1) (2010) 51–58.
[41] H.Y. Lin, W.G. Tzeng, A secure erasure code-based cloud storage system with secure data forwarding, IEEE Trans. Parallel Distrib. 23 (6) (2012) 995–1003.
[42] L.-J. Pang, Y.-M. Wang, A new (t, n) multi-secret sharing scheme based on Shamir's secret sharing, Appl. Math. Comput. 167 (2) (2005) 840–848.
[43] A. Parakh, S. Kak, Space efficient secret sharing for implicit data security, Inform. Sci. 181 (2) (2011) 335–341.
[44] A. Parakh, S. Kak, Online data storage using implicit security, Inform. Sci. 179 (19) (2009) 3323–3331.
[45] S. Ruj, A. Nayak, I. Stojmenovic, DACC: Distributed access control in clouds, in: Proc. 10th IEEE Int. Conf. on Trust, Security and Privacy in Computing and Communications, TrustCom 2011, 8th IEEE Int. Conf. on Embedded Software and Systems, ICESS 2011, 6th Int. Conf. on FCST 2011, 2011, pp. 91-98.
[46] B.K. Samanthula, Y. Elmehdwi, G. Howser, S. Madria, A secure data sharing and query processing framework via federation of cloud computing, Inform. Sci. 48 (2015) 196–212.
[47] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A.G. Dimakis, R. Vadali, S. Chen, D. Borthakur, XORing Elephants: Novel erasure codes for big data, in: 39th International Conference on Very Large Data Bases, VLDB, 2013, pp. 325–336.
[48] N.B. Shah, K.V. Rashmi, P.V. Kumar, K. Ramchandran, Interference alignment in regenerating codes for distributed storage: Necessity and code constructions, IEEE Trans. Inform. Theory 58 (4) (2012) 2134–2158.
[49] C. Wang, Q. Wang, K. Ren, N. Cao, W. Lou, Toward secure and dependable storage services in cloud computing, IEEE Trans. Serv. Comput. 5 (2) (2012) 220–232.
[50] J.J. Wylie, M.W. Bigrigg, J.D. Strunk, G.R. Ganger, H. Kiliccote, P.K. Khosla, Survivable information storage systems, Computer 33 (8) (2000) 61–68.
[51] C.C. Yang, T.Y. Chang, M.S. Hwang, A (t, n) multi-secret sharing scheme, Appl. Math. Comput. 151 (2) (2004) 483–490.
[52] S.J. Lin, W.H. Chung, Y.S. Han, Novel polynomial basis and its application to reed-solomon erasure codes, in: Ann. IEEE Symp. Found. 2014, pp. 316–325.
[53] C. Asmuth, J. Bloom, A modular approach to key safeguarding, IEEE Trans. Inform. Theory 29 (2) (1983) 208–210.
[54] M. Mignotte, How to share a secret, in: Workshop on Cryptography, 1982, pp. 371-375.
[55] P.M. Matutino, R. Chaves, L. Sousa, An efficient scalable RNS architecture for large dynamic ranges, J. Signal Process. Syst. 77 (1–2) (2014) 191–205.
[56] F. Barsi, P. Maestrini, Error detection and correction by product codes in residue number systems, IEEE Trans. Comput. 23 (9) (1974) 915–924.
[57] V.T. Goh, M.U. Siddiqi, Multiple error detection and correction based on redundant residue number systems, IEEE Trans. Commun. 56 (3) (2008) 325–330.
[58] N.I. Chervyakov, P.A. Lyakhov, M.G. Babenko, A.I. Garyanina, I.N. Lavrinenko, A.V. Lavrinenko, M.A. Deryabin, An efficient method of error correction in fault-tolerant modular neurocomputers, Neurocomputing 205 (2016) 32–44.
[59] S.R. Moosavi, T.N. Gia, E. Nigussie, A.M. Rahmani, S. Virtanen, H. Tenhunen, J. Isoaho, End-to-end security scheme for mobility enabled healthcare internet of things, Future Gener. Comput. Syst. 64 (2016) 108–124.
[60] M. Rahmani, T.N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, P. Liljeberg, Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach, Future Gener. Comput. Syst. (2017).
[61] S.E. Sarma, S.A. Weis, D.W. Engels, RFID systems and security and privacy implications a brief introduction to RFID systems, Security 82 (2002) 454–469.
[62] S.A. Weis, S.E. Sarma, R.L. Rivest, D.W. Engels, Security and privacy aspects of low-cost radio frequency identification systems, Lect. Notes Comput. Sci. 2802 (2004) 201–212.
[63] A. Khan, M.H. Rehmani, A. Rachedi, Cognitive-radio-based Internet of Things: applications, architectures, spectrum related functionalities, and future research directions, IEEE Wirel. Commun. 3 (24) (2017) 17–25.
[64] DDoS, 10 things to know about the October 21 IoT DDoS attacks, 2016. https://www.welivesecurity.com/2016/10/24/10-things-know-october-21-iot-ddos-attacks/79.(Accessed 21 June 2017).
[65] A. Hodjat, I. Verbauwhede, Minimum area cost for a 30 to 70 Gbits/s AES processor, in: IEEE Comp. Soc. Ann. 2004, pp. 83-88.
[66] A. Moradi, A. Poschmann, S. Ling, C. Paar, H. Wang, Pushing the limits: A very compact and a threshold implementation of AES, Lect. Notes Comput. Sci. 6632 (2011) 69–88.
[67] E. Käsper, P. Schwabe, Faster and timing-attack resistant AES-GCM, Lect. Notes Comput. Sci. 5747 (2009) 1–17.
[68] B. Preneel, Perspectives on Lightweight Cryptography. http://homes.esat.kuleuven.be/~preneel/preneel_lightweight_shanghaiv1.pdf. (Accessed 21 June 2017).
[69] X. Wang, H. Yu, How to break MD5 and other hash functions, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2005, pp. 19–35.
[70] P. Pritzker, P.D. Gallagher, SHA-3 standard: Permutation-based hash and extendable-output functions, in:. Inf. Tech. Lab. Natl. Inst. Stand. Technol. 2014, pp. 1–35.
[71] L. Munson, Greatfire.org faces daily $30,000 bill from DDoS attack, 2015. https://nakedsecurity.sophos.com/ru/2015/03/20/greatfire-org-faces-daily-30000-bill-from-ddos-attack/. (Accessed 21 June 2017).
[72] K. Leswing, A massive cyberattack knocked out major websites across the internet, 2016. http://www.businessinsider.com/amazon-spotify-twitter-github-and-etsy-down-in-apparent-dns-attack-2016-10/.(Accessed 21 June 2017).
[73] Kaspersky lab, Kaspersky DDoS Intelligence Report for Q1 2016. https://securelist.com/analysis/quarterly-malware-reports/74550/kaspersky-ddos-intelligence-report-for-q1-2016/ . (Accessed 21 June 2017).
[74] D. Gage, Nirvanix Files for Chapter 11 Bankruptcy, 2013. http://blogs.wsj.com/venturecapital/2013/10/01/nirvanix-files-for-chapter-11-bankruptcy/ (Accessed 21 June 2017).
[75] WCO, The worst cloud outages of 2014 (so far), 2014. http://www.infoworld.com/article/2606209/cloud-computing/162288-The-worst-cloud-outages-of-2014-so-far.html/.(Accessed 21 June 2017).
[76] S. Wu, K.-C. Li, B. Mao, M. Liao, DAC: Improving storage availability with deduplication-assisted cloud-of-clouds, Future Gener. Comput. Syst. 74 (2017) 190–198.
[77] Geekbench Browser. https://browser.primatelabs.com. (Accessed 21 June 2017).
[78] M. Grangetto, P. Cosman, G. Olmo, Joint source/channel coding and MAP decoding of arithmetic codes, IEEE Trans. Commun. 53 (6) (2005) 1007–1016.
[79] N.I. Chervyakov, M.G. Babenko, P.A. Lyakhov, I.N. Lavrinenko, An approximate method for comparing modular numbers and its application to the division of numbers in residue number systems, Cyber. Syst. Anal.+ 50 (6) (2014) 977–984.
[80] M. Barzu, F.L. Tiplea, C.C. Drăgan, Compact sequences of co-primes and their applications to the security of CRT-based threshold schemes, Inform. Sci. 240 (2013) 161–172.

Nikolay Chervyakov received the Ph.D. degree from Stavropol State University (SSU) in 1972, and doctoral degree in 1987. He is a full professor since 1989, and the head of Department of Applied Mathematics and Computer Science of SSU since 2004. He is author of over 700 scientific publications and over 100 patents. He is a scientific advisor of over 70 doctoral and Ph.D. dissertations. His main interests include algebraic structures over Galois field, modular arithmetic, artificial neural networks computing, cloud computing, digital signal processing and cryptography.

Please cite this article in press as: N. Chervyakov, et al., AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security, Future Generation Computer Systems (2017), https://doi.org/10.1016/j.future.2017.09.061.

Mikhail Babenko graduated from Stavropol State University (SSU) in 2007 with degree in mathematics. Received Ph.D. degree in mathematics from SSU in 2011. He works as assistant professor in Department of Applied Mathematics and Mathematical Modeling since 2012. He is an author of over 63 publications and 5 patents. His research interests include cloud computing, high-performance computing, residue number systems, neural networks, cryptography.

Andrei Tchernykh received his Ph.D. degree from Institute of Precise Mechanics and Computer Technology of the Russian Academy of Sciences, Russia in 1986. He is holding a full professor position in Computer Science Department at CICESE Research Center, Ensenada, Baja California, Mexico, and a head of Parallel Computing Laboratory. He is a member of the National System of Researchers of Mexico (SNI), Level II, and a founding member of the Mexican Supercomputer Society. He has published more than 200 papers in refereed journals and conferences, and served as a TPC member and general co-chair of more than 240 professional peer reviewed conferences. He was invited as a visiting researcher at prestigious universities and research centers. He leads a number of research projects and grants in different countries. He has served as a member of the editorial boards and guest editor of several scientific journals. His main interests include resource optimization technique, adaptive resource provisioning, multi-objective optimization, computational intelligence, incomplete information processing, cloud computing and security.

Nikolay Kucherov received Bachelor degree in mathematics from Stavropol State University in 2012, and Master degree in parallel technologies from NCFU in 2014. He is postgraduate student and junior researcher at NCFU since 2014. His research interests include cloud computing, modular arithmetic, residue number systems, FPGA, threshold cryptography, high performance computing.

Vanessa Miranda-López received a Bachelor degree in electronics engineering from Technological Institute of Sonora, Mexico in 2006, and Master degree in computer sciences from CICESE Research Center in 2010. Her interests include cloud computing, grid scheduling, big data, security and electronic design.

Jorge M. Cortés-Mendoza received his Bachelor degree in computer sciences from the Autonomous University of Puebla (Benemrita Univercidad Autnoma de Puebla, Mexico) in July 2008, and his Master degree in computer science from CICESE Research Center in March 2011. Since 2013, he is working on distributed computing, cloud computing, load balancing and scheduling.

Please cite this article in press as: N. Chervyakov, et al., AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security, Future Generation Computer Systems (2017), https://doi.org/10.1016/j.future.2017.09.061.