Journal of Electrical Engineering and Information Technologies, Vol. **4,** No. 1–2, pp. 51–61 (2019)
Article 161 ISSN 2545–4250
Received: October 17, 2019 e: ISSN 2545–4269
Accepted: November 5, 2019 UDC: 004.33.056.3:004.421.056.56]:004.6

Original scientific paper

SECURITY IN DISTRIBUTED STORAGE SYSTEMS – ENCRYPTION ALGORITHM VS. CODING SCHEMES

Nataša Paunkoska (Dimoska), Aleksandar Risteski

Faculty of Electrical Engineering and Information Technologies,
"Ss. Cyril and Methodius" University in Skopje,
Rugjer Bošković bb, P.O. box 574, 1001 Skopje, North Macedonia
natasa.paunkoska@gmail.com

A b s t r a c t: Distributed storage systems (DSS) are crucial for managing large amount of data produced on daily bases. The concept for data distribution and storage play important role for obtaining efficient, reliable and secure system. Currently used replication technique is almost replaced with use of other coding schemes, which reduces the unnecessary system redundancy, improve the reconstruction and repair processes and increases the security. In this paper we are considering the DSS security in sense of using certain methods for preventing possible attacks. A security model is conceived by predicting all possible weak points in the DSS system, where potential attacks can happen. Along with the proposed model is made a comprehensive overview of literature that covers applying different techniques for handling those issues. Hence, a classification is prepared of all used methods, encryption algorithms and coding schemes, with the appropriate concepts resolving various type of attacks. Further, system performance analyses are given and is concluded which method performs better when the system considers the security issue and which one when it is excluded.

Key words: distributed storage system; security; coding schemes; encryption algorithm; attacks

СИГУРНОСТ ВО ДИСТРИБУИРАНИ СИСТЕМИ ЗА СКЛАДИРАЊЕ – ЕНКРИПЦИСКИ АЛГОРИТМИ НАСПРОТИ КОДНИ ШЕМИ

А п с т р а к т: Дистрибуираните системи за складирање (ДСС) се клучни во управувањето со податоци од големи размери кои се создаваат на дневно ниво. За да се постигне ефикасен, доверлив и сигурен систем, важна улога има начинот на дистрибуција и складирање на податоците. Техниката на репликации, која моментално се користи, веќе се заменува со користење на други кодни техники кои имаат за цел да ја намалат непотребната редундантност во системот, да ги подобрат процесите на реконструкција и поправка и да ја зголемат сигурноста. Во овој труд акцентот се става на сигурноста во ДСС, во смисла на користење одредени методи за спречување потенцијални напади. Сигурносниот модел е конструиран врз претпоставка на одредување сите потенцијални слаби точки во ДСС каде што може да се случи напад. Заедно со предложениот модел е направена детална анализа на литература што опфаќа истражување за справување со тие предизици. Дополнително е подготвена класификација на сите користени методи, енкрипциски алгоритми и кодни шеми, според концептот што тие го користат и со кој тип на напад се справуваат. Направена е и анализа на перформансите на системот и е заклучено кој метод е најдобар кога се зема предвид сигурноста, а кој кога таа се исклучува.

Клучни зборови: дистибуирани системи за складирање; сигурност; кодни шеми; енкрипциски алгоритми, напади

1. INTRODUCTION

A distributed storage system (DSS) is a widely used technology for data storing in a efficient and reliable way. All distributed applications usually relies on this technology, because for them it represents essential building block. These systems consist of a collection of n storage nodes (servers) that are physically dispersed interconnected. In such environment the nodes may be individually unreliable,

which by applying additional redundancy the system as a whole can become reliable. Most of the DSSs these days uses three time replication for each part of the data that going be stored in the system. This is not very suitable solution, because of the unnecessary overwhelming system storage. Another way to be ensure the reliability and to be reduce the bandwidth required for repair (process when some node fails and lost what is stored on it, then new node comes into play and recovers what was lost by help and information from the other alive nodes) can be provided by using coding schemes for data distribution. In particular, linear network coding has turned out to offer good performance both in theory and in practice. Additionally, applying code schemes for data storage in DSS archives certain information-theoretic secrecy.

Data stored on the nodes belongs to various type and nature. Lately, there is a rapid increase for storing sensitive type of data, such as health care records, customer records or financial data. Finding a way to protect such kind of data while there are in transit, as well as, while are at rest is crucial. In the traditional networks exist a large variety of sophisticated protocols for secure transfer of data over insecure networks, like SSH [7], TLS [8], IPSec [9]. Although those protocols work well, most successful attacks that result in disclosure of confidential data don't occur while data is transmitted over a network. Most often a much easier target are data on a mass storage devices. Getting access to such devices through social engineering, weak passwords or security holes in the access control system is a relatively simple task compared to successfully intercepting and interpreting a network communication.

Therefore, a successful task would be designing a coding schemes for data distribution/storage in DSS that will enable data security against different types of attacks, while simultaneously keep the system reliable, resilient to node failures and efficient in the node repair mechanism. Until now, different coding schemes are adjusted and designed for proper and effective working of DSS systems. Some of them offer more efficient distribution process by trading some of the other parameters, some increase the bandwidth needed for the repair process, increases the security, etc.

Relying only on coding schemes in DSS for achieving the security is not wise. This approach has some limitations. Usually, coding schemes can protect the data, if the intruder compromise less number of nodes than the number determined by the used distribution code. Therefore, the researchers try to include encryption algorithms in the DSSs schemes to deal with the security threat issue. Adding encryption is important for ensuring the confidentiality of the data. However, traditional cryptographic primitives are inadequate for network coding which requires that data packets from different nodes can be combined according to the coding scheme. An optimally secure distributed storage architecture would minimize the use of cryptographic operations and avoid unnecessary decryption and re-encryption of data as long as the data does not leave the file system.

The goal of this paper is to appoint which are the vulnerable points in the distributed storage system with respect of security attacks. Following the proposed threat model is provided an overview of existing literature dealing with this problem. The revision of the survey shows that the problem is generally examined from two points of view: applying specific coding scheme for data storage that will offer certain information-theoretic secrecy and coding scheme plus encryption algorithm for providing additional security. Encryption algorithms usually considered the key management approach for doing the encryption process. Based on all possible models for ensuring security, is provided a distinction for which mechanism is most suitable for dealing of which type of attack within the DSS frameworks.

The paper is organized as follows: Section 2 gives how is defined the DSS system, section 3 explains the security model with all possible attacks, section 4 provides a related work with all developed concepts for dealing the security issues in such system, section 5 gives discussion for which approach are appropriate for which attack and system performance analysis. Section 6 concludes the paper.

2. DSS SYSTEM

Distributed storage system (DSS) is dispersed network consist of n nodes (servers). On the nodes a user data is stored with help of the data collector (DC), which distributes the user messages in some predefined manner (scheme). Classical DSS system uses the principle of replication to distribute and store the data over the network. Usually, replication schemes enables three copies of each message, which are stored on different servers. Those replicas serves as a back-up in case of some failures in the system. This is noneffective way for data storing due to the unnecessary added redundancy. Replacing the replication with traditional coding schemes enables improved system performance [1]. Therefore, in our paper we will concentrate on using the

concept of coding schemes for data distribution instead of replication. Some methods for using coding scheme in DSS can be found in [2 – 6].

The model uses the (n, k, d) approach. This mean that the DSS is consist of n number of nodes and the user message B will be deployed on them. The message B is broke on $k(k < n)$ parts, leading to $\frac{B}{k} = \alpha$, and these pieces of quantity information α will be stored on k nodes. Then $(n - k)$ redundancy pieces will be provided by making combination of the previous k, and those parts will be stored on the remaining $(n - k)$ nodes in the system.

If the user wants to retrieve the message back, then the DC contacts any k nodes in the system, downloads what is storeds on them and reconstructs the original message before it sends back to the user. The amount of information that will be downloaded is $k\alpha$. This procedure is known as reconstruction process. In the system can occur some damage or failure of some of the servers, and the information stored on them to be lost. In this case new node (newcomer) will be added in the system. The newcomer needs to retake all function of the failed node. Therefore, it contacts any $d(k \leq d < n)$ nodes in the system, downloads from them $\beta(\beta < \alpha)$ amount of information, makes some computation and repair what was lost of the failed node. This process is known as repair process.

Dimakis et al. in [2] introduces correlation between the total downloaded amount $d\beta = \gamma$, known as repair and the storage α. Based on that tradeoff, two extreme points are obtained, Minimum Storage Regeneration (MSR) and Minimum Bandwidth Regeneration (MBR). Further, almost all code constructions in this research area follows this concept. The analysis in this work also will be provided with respect of this concept.

3. SECURITY MODEL WITHIN DSS

Securing the data that is distributed, stored and maintained, in the DSS system is of essential meaning for gaining the user trust. To deal successfully with this issue, it is important to be distinguished which are the possible weak points in the system, i.e., the diverse attacks that can be performed on the data.

Take into consideration a general (n, k, d) DSS system, the secure model will differentiate two common types of attacks: passive and active. When passive attacks occur, then the attacker only eavesdrop the data without any changes in the content. Unlike the passive one, the active attack except the eavesdropping the intruder can modified the original data. Under these two general separations, further, it can be distinguished after three various attacks, the division is given in Figure 1.

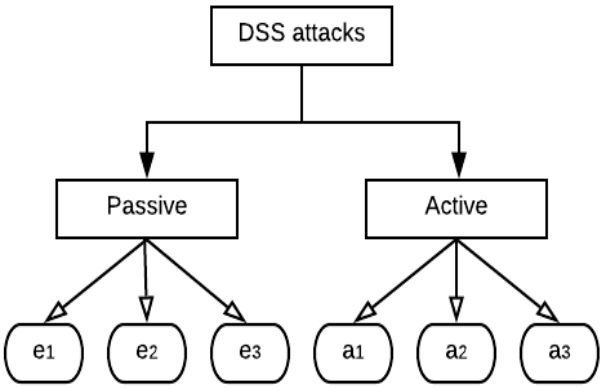

Fig. 1. Types of attacks that are possible to happen in the DSS system

The proposed security model that has an aim to protect the data in the DSS is given in Figure 2. In the figure are depicted all possible attacks that can happen in the system, hence, it is shown where is necessary to be prevented.

The eavesdropper (passive attack) in the DSS can observes data that can be found on three 'weak' system points:

• 'e_1' - the eavesdropper observes data stored on some nodes in the systems. When using only coding scheme for preserving the security the maximum number of nodes that can be affected of this type of attack is $l(l < k)$, because k is the number of nodes needed for reconstructing the original stored message. Each node stores α amount of information. Adding encryption algorithm increases the possibility the number of l to be larger, but also requires increased computational power in the system.

• 'e_2' – the eavesdropper observes the data needed for performing the repair process. When a node fails, the data stored on it is lost. A newcomer or new node is added in the system, contacts $d(k \leq d < n)$ nodes downloads form them $d\beta$ quantity of information, performs some computation and recovers what is lost. The potential attack is the newcomer to be compromised or to be the attacker.

• 'e_3' – the attacker eavesdropped the connection between the nodes. The attacker can listens what is transfer from the Data Collector (DC) (responsible entity for data distribution from the user to the system, and vice versa to collect the data from the stored nodes and return back to the users) to the DSS nodes and between the newcomer and the other nodes.

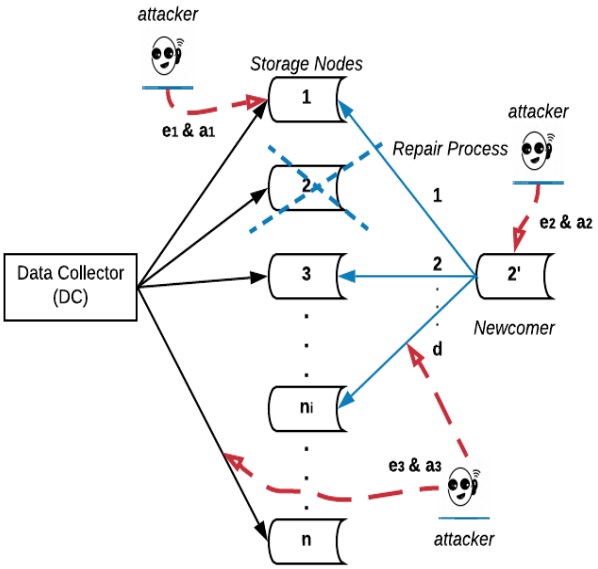

Fig. 2: DSS security model

The intruder during the active attack can reach to the same spots in the DSS system as in the passive attack case, and change some symbols in the content:

• 'a_1' – the intruder can access the data stored on some DSS nodes and eventually modifed one or couple symbols of the content. The active attack is in close relationship with the passive one. If some is able to modified part of the information, then it is able to eavesdrop too. Therefore, we must assume that the intruder can affect no more than $l(l < k)$ nodes or no more than $l\alpha$ symbols can be changed, if information-theoretic secrecy is only consider. And more of encryption is added.

• 'a_2' – during the repair process the intruder is the newcomer and can modify the repair content. The repair content is recovered after contacting d alive helper nodes in the system. One or more symbols from the $d\beta$ quantity that is recovered can be than changed.

• 'a_3' – the intruder can somehow interfere on the connection where the data is transfer, like in the traditional network communication, and to contribute in changing part of the data. The connections that are affected from this type of attack are the same ones as in the attack e_3.

4. RELATED WORK

Security in DSS can be obtained by using a specific coding scheme for distribution that introduces information-theoretic secrecy, or by using encryption algorithm combined with some coding scheme. A good survey of the security services provided by the existing storage systems can be found in [10, 11] where are addressed important security issues related to the data storage. In the section 4.1. we will give an overview of the existing works regarding this security issue in DSSs. The overview will contain only work after the time the paper [10] is published.

4.1. *DSS security managed by encryption*

Most of the research work that uses encryption deals the integrity and security of the data stored on the nodes. That means e_1 and a_1 type of attacks. Papers [12-14] takes into account only the passive attack e_1. The security in [12, 13] is achieved by using a key management mechanism for encryption and in [14] by uses specific scheme along with the distribution coding scheme. More precisely, in [12] the authors examine the problem of encrypted data storage in a distributed or grid computing environment, where storage capacity and data are shared across organizational boundaries. The architecture that is proposed uses decryption keys and their access is granted based on the grids data access permissions. The concept allows the storage servers not to be trusted, because of the existence of the access control models. Hence, in [13] the authors consider DSS where the storage nodes do not see the original encrypted data but only linear combinations. They have achieved to enable the user to remotely change the encryption key of a file stored in a DSS by safely delegating the re-encryption process to the storage network. The solution counts a key-homomorphic pseudorandom function (KH-PRF) in counter mode encryption. This scheme is applicable also when the data is directly stored into the system, i.e., normal

storage where the data is directly stored in the storage nodes without coding. The change of the encryption key prevents from the server maliciously to cooperate with other users to decrypt the data and weak access control of the systems that allow malicious users to obtain stored data. The scheme for security in [14] is called Private information retrieval (PIR) and aims to protect users from surveillance and monitoring, i.e. file can be downloaded without revealing any information of which file is actually downloaded to the servers storing it. Hence, communication price of privacy (cPoP) represent the lowest possible amount of downloaded data per unit of stored data, when there is no colluding nodes or a single spy node. Here, the authors propose new PIR scheme for DSS where data is stored using an arbitrary linear systematic code of rate $R > \frac{1}{2}$. Simultaneously, in such circumstances they achieved to optimize the cPoP of the protocol for several various coding schemes.

Papers [15–17] except considering only the passive attack e_1, they considering also the active type of attack a_1. In [15] the authors consider the confidentiality of network coding and, in particular, distributed storage systems in a setting where the adversary has complete control of the nodes but is computationally bounded. The authors propose scheme based on linear error correcting code using symmetric additively homomorphic encryption technique that is compatible with the linear network coding. The advantages achieved compared to ordinary encryption are: linear network coding can be applied if working directly with the plaintext messages, linear operations on the ciphertext space transfer to the plaintext space upon decryption; the encrypted parts of the file do not disclose which part is which, the part information can be kept in the plaintext domain. It makes it impossible for the storage nodes or the adversary to eavesdrop on which subsets of the data the user requests; the plaintext data can be first authenticated and then encrypted, for storage systems this ordering is often desirable to ensure plaintext integrity; and last the scheme provides simultaneous encryption and error correction.

The authors in [16] proposed a simple linear hashing scheme to detect errors in the storage nodes, when the error-correction capabilities that are built into the existing redundancy of the system are used. The main idea is constructing small projections of the data that preserve the errors with high probability and build on a pseudorandom generator that fools linear functions. That is applicable by using N-extended version of a code and creating a linear projection (hash) of each row on the same random vector. The key observation is that if the same random projection is used, this creates an error-correcting code for the hashes which can be communicated to the verifier. The benefit is that each hash has size only 1/N of the data in each row reducing the amount of communication to the verifier. The goal of the verifier that overlooks the state of the whole system is first to find the erroneous disk with the minimum data exchange and second to repair it by using the information stored on the other disks. Paper [17] to achieve the security gives a generic design for cryptographic file systems and its realization in a distributed storage-area network (SAN) file system. The principle is the key management is integrated with the meta-data service of the SAN file system. The implementation supports file encryption and integrity protection through hash trees.

Another issue where the security should be consider is the communication within the DSS system. Distributed systems require the ability to communicate securely with other computers in the network e_3 and a_3 type of attacks. To accomplish this, author in paper [18] applies a method for efficient key management inside a distributed system that uses identity based encryption (IBE). Public resources in a network are addressable by unique identifiers. Using this identifier as a public key, other entities are able to securely access that resource. This system allows secure, efficient, authenticated communication inside a distributed system.

4.2. *DSS security managed by coding schemes*

Passive attacks e_1 and e_2 are considered in the papers [19–23]. The authors in [19] provide explicit constructions of regenerating codes that achieve information-theoretic secrecy capacity in DSS. As a background they are using the regenerating code construction given in [6]. They consider a threat model where an eavesdropper may gain access to the data stored in a subset of the storage nodes, and possibly also, to the data downloaded during repair of some nodes. As an achievement, they set a secrecy bounds in different scenarios of the defined threat model and simultaneously enables reliability, availability, and efficient node repair in the DSS system. Paper [20] studied the same problem of securing data in distributed storage systems against eavesdropping. The difference form [19] is that here the focus is put on systems that implement linear codes and exact repair. The authors determined the maximum file size that can be stored securely in

these systems for any number of compromised nodes, when the repair degree is $d = n - 1$. In [21] the authors applied in the DSS a coding scheme that uses the interference alignment concept. This approach allows efficient repair process and gaining security against the passive attacks e_1 and e_2. Rawat et al. in [22] study the local-repairable codes applied in DSS and exams the security achieved with their implementation. The security is investigated in the presence of colluding eavesdroppers, where eavesdroppers are assumed to work together in decoding stored information. The proposed secure scheme offer an improved bound on the secrecy capacity for minimum storage regenerating codes given in [19], a new bound on minimum distance for locally repairable codes, code construction for locally repairable codes that attain the minimum distance bound, and repair-bandwidth-efficient locally repairable codes with and without security constraints.

Additionally, except the passive attacks the active ones a_1 and a_2 are discussed in [23–25]. In [23] are define the secrecy and resiliency capacities of a distributed storage system as the maximum amount of information that it can store safely, respectively, in the presence of an eavesdropper or a malicious adversary. Here, except passive attack an active one is also considered. The secrecy bounds that are obtained are set up over DSS code construction using on one side the repetition code and on another side on the MDS code. In continuation, the same authors in their next paper [24] concentrates only on active attacks in DSS systems. To deal the secrecy issue, they provide explicit linear DSS code constructions for data distribution. The secrecy bound which is provided covers only the case where the intruder can attack in the process of repair data. Hence, this concept offers a way to shortlist the malicious nodes and expurgate the system. Paper [25] deals also e_1, e_2, a_1 and a_2 types of attacks. As a coding scheme is used the MDS twin-code framework that ensures the security against passive attacks. Hence, by using additional hash function is provide security against active attacks.

Weak information-theoretic secrecy is another way to secure a distributed storage system. The goal is to construct a weakly secure DSS that leaks no meaningful information to the attaker. This kind of secrecy deals the passive attack e_1 and is elaborated in [26–28]. Paper [26] uses the code scheme Product-Matrix Minimum Bandwidth Regenerating plus coset codes to achieve the security. A coding scheme that uses the interference alignment concept plus coset codes is used in [27] and shortened MSR codes with coset in [28].

In [29] the key technical contribution is in developing novel information theoretic converse proofs for the Type-II adversarial scenario, i.e., adversary that can observe the repair data. It is shown that in the presence of Type-II attacks, the only efficient point in the storage-vs-exact-repair-bandwidth tradeoff is the MBR (minimum bandwidth regenerating) point. This is in sharp contrast to the case of a Type-I attack, adversary which can wiretap the data stored on the nodes, in which the storage-vs-exactrepair-bandwidth tradeoff allows a spectrum of operating points beyond the MBR point. The data distribution in the system is done using the regenerating codes.

5. DISCUSSION

This section provides classification of all used methods for obtaining security, i.e. encryption algorithms and coding schemes for achieving information-theoretic secrecy. The division is based on which types of attack are resolved by which security method. Hence, system performance analyses are done regarding the amount of data that can be stored, first without considering the security and then adding the security parameter.

5.1. *Classification of used methods*

Distributed storage systems are very popular these days for storing and maintaining enormous quantity of data, like 'Big Data'. Various type of information are sent to be kept in these systems by the users. Very often, important and sensitive data are put there with hope that they will be stored in a safe way and untouched. The DSS represent large network of dispersed servers (nodes) connected between each other. This means that such environment is not very truthful and reliable. All servers and connection are potential points of attacks. The vulnerable points are explained in section 3.

Security in DSS generally can be achieved by using a concrete coding scheme that introduces information-theoretic secrecy or by using some encryption algorithm. As is discussed in section 2, a good way for data distribution is by using some code scheme. Why? Because it reduces the unnecessary storage redundancy, and is very efficient in performing data reconstruction of the original message and repair of data lost due to some server

damage. An opportunistic thing is that the same coding scheme can offer certain data information-theoretic secrecy. Therefore, many researches tries to adjust or design a coding scheme that simultaneously can enable efficient, reliable and secure DSS. Of course, not all parameters can be satisfied equally, there will be always some trading. So, different coding scheme provides different outputs. If we talk about ensuring the security, some research works concentrate on dealing with only one type of attack.

In the review above we can see that most of the works are directed to find a way how to protected the data stored on the nodes, in seanse of not to be observed or modified. The majority of them, considers the repair process too, i.e. care that the new node added in the network maybe is compromised. One of the solutions is to limited the amount of information that the intruder needs for recovering the lost data, what prevents not to reveal the whole original message. What is interesting here to be mentioned, that there is no work that considers the connections between the nodes and provides information-theoretic secrecy in DSS, i,e. secure the links where the data is transfer.

On other side, using only code schemes for achieving secrecy has some limitations. This approach can not fully preserve the data security and integration. For example, all DSS code schemes of shape (n, k, d) assumes that the number of compromised nodes in the system l is not greater or equal to k. k is the number of nodes that need to be contacted for reconstructing the entire original message. Having $l \geq k$, we can say that the intruder can obtain the user message. This issue the researchers try to handle by applying some encryption mechanisms. Because of the enormous stored data and the offent request-responses that happen in the system, the DSS designer must take care of encryption complexity. An optimally secure distributed storage architecture should minimize the use of cryptographic operations and avoid unnecessary decryption and re-encryption of data as long as the data does not leave the file system. Generally, the offered encryption algorithms are based on method for key management to practically and efficiently secure internal communication. Hence, when using cryptography to provide secrecy, the legitimate parties usually hold a common secret key which is required to decrypt the stored information. In this case, unlike securing with coding schemes that usually protects data stored on nodes and data provided during the repair process, encryption protects data stored on nodes and the connections within the DSS system.

The classification of which secure methods are used within the paper works and with what kind of attacks are dealing is given in Table 1.

5.2. *Performance Analyses*

In this subsection we are comparing different coding schemes regarding the data that can be distributed in the DSS systems, i.e. storage capacity, and analysis how they are performing. Following section 2, we are grouping the codes in to ones constructed to achieve the MBR point, and the others to achieve the extreme MSR point. The first part of the analysis is done without taking the care of the security parameter. In Table 2 are listed all bounds or the measurements of the quantity of data that can be distributed in the distributed storage systems when different coding schemes are used.

Figure 3 considers eight different coding schemes based on MSR point. For all of them general code for performing is chosen

$$(n,k,d,\alpha,\beta) = (15,k,12,6,1) \text{ and } k = 3,\ldots,11.$$

From here we can conclude that depending of the different value of k, the number of nodes needed for reconstruction, varies which code performs better. In case of small k, MSR and IA codes are the best, but for large k, TwinMSR and PM-MSR outperform.

Next comparison, Figure 4, analyzes five different coding schemes based on MBR point. The general used code is $(n,k,d,\alpha,\beta) = (15,k,12,6,1)$ and $k = 3,\ldots,11$. In this case the MBR code gives the best result for all values of k. For large k TwinMBR also give good storage capacity performance.

Storage capacity is one parameter for analysis. In DSS other relevant system parameters are repair bandwidth, reconstruction capacity, simultaneous repair of failed nodes. Analysis can be done also regarding these values. The results will vary for which code is better. In order to be selected the best one, some trade off must be made, based on what we want to achieve.

The second part overviews the data distribution in DSS, while the security is taken into consideration. Generally, if we compare the amount of information that can be stored in DSS with security, we can notice that the quantity is a way smaller than the one stored without security. This is expected, due to the assumption that the intruder can corrupt some part of the message, that is excluded from the entire data, and the rest will be secure because won't be enough for obtaining the original one. This is the main concept in the information-theoretic secrecy.

Table 1

Used concepts in the papers for dealing the security attacks

Paper	Attack	Coding scheme	Encryption algorithm
[12]	e_1		Decryption keys based on grid data access permissions
[13]	e_1		Key-homomorphic pseudorandom function (KH-PRF)
[14]	e_1		Private information retrieval (PIR)
[19]	e_1,e_2	Regenerating codes	
[6]	e_1,e_2	Product-matrix code	
[20]	e_1,e_2	Linear codes and exact repair	
[21]	e_1,e_2	Interference alignment concept	
[22]	e_1,e_2	Local-repairable codes	
[15]	e_1,a_1		Symmetric additively homomorphic encryption
[16]	e_1,a_1		Linear hashing scheme
[17]	e_1,a_1		Key management with the meta-data service
[23]	e_1,a_1,e_2,a_2	MDS code and repetition	
[24]	e_1,a_1,e_2,a_2	explicit linear DSS code	
[25]	e_1,a_1,e_2,a_2	MDS twin-code	
[26]	e_1	Product-matrix plus coset	
[27]	e_1	Interference alignment plus coset	
[28]	e_1	MDS twin-code	
[18]	e_3, a_3		Identity based encryption (IBE)
[29]	e_2,a_2	Minimum bandwidth regenerating	

Table 2

Comparison of amount of data that can be stored in DSSs, when the security is not consider and then the security is included

Coding Schemes	Data distribution without security	Data distribution with security
MBR regenerating codes	$B = \left(kd - \binom{k}{2}\right)\beta, \alpha = d\beta$	$B_s = \left(kd - \binom{k}{2}\right)\beta - \left(ld - \binom{l}{2}\right)\beta$
MSR regenerating codes	$B = k\alpha, d\beta = \alpha + (k+1)\beta$	$B_s = (k-l)\alpha$
MBR MDS Twin	$B = k(k+1)$	$B_s = k(k-l)$
MSR MDS Twin	$B = k^2$	$B_s = k(k-l-l')$
Linear codes and exact repair	$B = k\alpha$	$B_s = (k-l-l')\left(1 - \frac{1}{d-k+1}\right)l'$
MBR product matrix	$B = k(d-k) + \frac{k(k+1)}{2}$	$B_s = \left(kd - \binom{k}{2}\right)\beta - \left(ld - \binom{l}{2}\right)\beta$
MSR product matrix	$B = \alpha(\alpha+1), \alpha = k-1, d = 2\alpha$	$B_s = (k-l)(\alpha - l'\beta)$
Interference alignment concept	$B = k(d-k+1)\beta$	$B_s = (k-l-l')(\alpha - l')$
Local-repairable codes	$B = \left(\left(kn - k^2 + k - t\left\lceil\frac{k}{r}\right\rceil - 1\right) - \binom{k}{2}\right)\beta$	$B_s = (k-l-l')(\alpha-\beta)$
MSR product-matrix plus coset	$B = \alpha(\alpha+1)$	$B_s = B - 1$
MBR product-matrix plus coset	$B = k(d-k) + \frac{k(k+1)}{2}$	$B_s = B - 2$
Interference alignment plus coset	$B = k\alpha$	$B_s = k\alpha - 2$
Minimum bandwidth regenerating - Tandon	$B = k(d-k) + \frac{k(k+1)}{2}$	$B_s = B\frac{\alpha}{n-1}$

To analysis data storage with secrecy, we must assume that there are intruders in the system. Following section 3, all potential points of attack are known. Meaning for e_1 and a_1 types of attack, the intruder can corrupt the stored data on any node. We will note by l the number of corrupted nodes. Only limitation is $l < k$, because k is number of nodes needed for recovering the original message. For the types of attack e_2 and a_2, when some node fails and new node (newcomer) is added in the system, the intruder is the newcomer and that node is corrupted. We will note these nodes by $l^{'}$. The limitation is the same $l^{'} < k$. If we want to combine those types of attack, respectively the condition $l + l^{'} < k$ need to be satisfy.

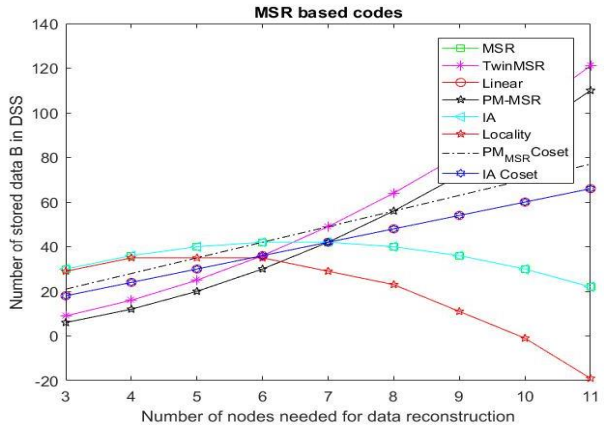

Fig. 3: Storage capacity, i.e., number of symbols that can be stored in DSS without considering the security. Code constructions based on MSR are consider for the concrete parameters $(n,k,d,\alpha,\beta) = (15,k,12,6,1)$ and $k = 3,\ldots,11$.

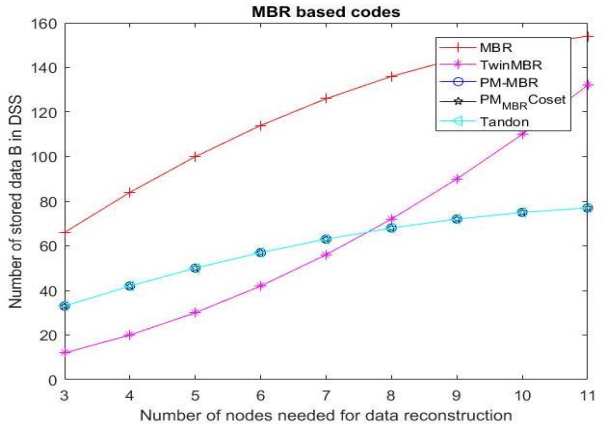

Fig. 4: Storage capacity, i.e. number of symbols that can be stored in DSS without considering the security. Code constructions based on MBR are consider for the concrete parameters (n, k, d, α, β) = (15, k, 12, 6, 2) and k = 3…., 11.

In Figure 5 are considered all coding schemes that deals with e_1 and a_1 type of attacks. The measurement of their comparison is amount of information that can be stored securely in the DSS. There are nine such different coding schemes. The general code that is selected for their simulation is $(n,k,d,\alpha,\beta) = (15,9,14,6,1)$ and the number of corrupted node is changeable starting from $l = 3,\ldots,8$. From the result we can notice that all schemes as the number of corrupted nodes is increasing, the secure storage capacity is decreasing. The best performance has TwinMBR, then MBR, PM-MBR and the worst performs IA. Considering the comparison graphs without security constrain, IA for small k performs very good, and TwinMBR for small k very bad.

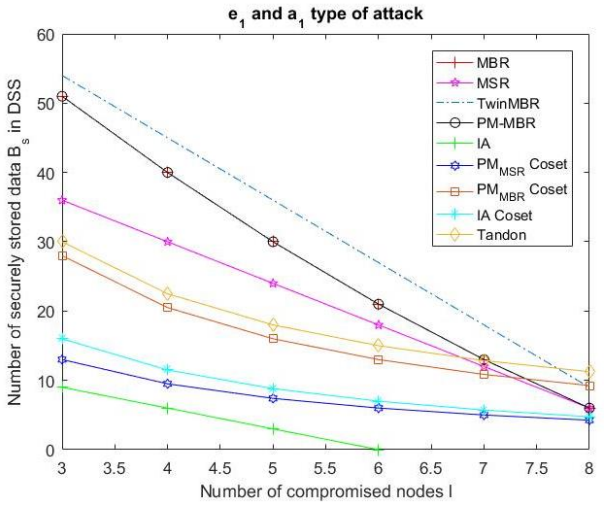

Fig. 5: Secure storage capacity, i.e. number of symbols that can be securely stored in DSS. Code constructions considering e_1 and a_1 attacks for the concrete parameters (n, k, d, α, β) = (15, 9, 14, 6, 1) and number of compromised nodes $l = 3,\ldots,8$.

Coding schemes that deals with e_2 and a_2 type of attacks are explored in Figure 6. There are four such coding schemes that follows the general code for simulation $(n,k,d,\alpha,\beta) = (15,9,14,6,1)$, $l = 1$ number of corrupted nodes, where the intruder observed the stored data, and $l^{'} = 3,\ldots,8$ number of corrupted newcomers. In this situation, same as previous, the secure storage capacity decreases as the number of intruders increases. TwinMSR and PM-MSR performs better in this case, and the Linear code is weaker. Considering the comparison graphs without security constrain, Linear code is somewhere in middle, and TwinMSR and PM_MSR are good only for large k.

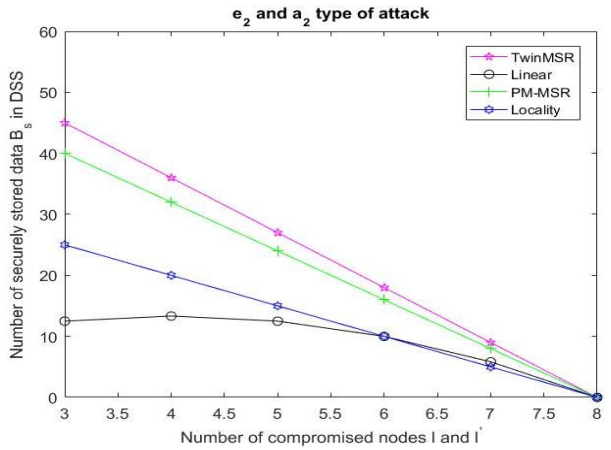

Fig. 6. Secure storage capacity, i.e. number of symbols that can be securely stored in DSS. Code constructions considering e_2 and a_2 attacks for the concrete parameters (n, k, d, α, β) = (15, 9, 12, 6, 1) and number of compromised nodes $l = 1$ and $l' = 3,\ldots,8$.

6. CONCLUSION

Managing and storing the Big Data produced in a daily basis is a challenge. Distributed storage systems (DSS) are such systems that deal the issue of data storage. Properly distributing the data on the servers geographically spread across the network is crucial. Therefore, various code constructions are proposed that need to satisfy some properties of the DSS, like efficiency, reliability, reconstruction process, repair process, data security. This paper provides comprehensive overview of all used methods for achieving the above mention goal, with special concern on the security issue. Different coding techniques are discussed that can provide information-theoretic secrecy and simultaneously satisfy some of the DSS properties. Moreover, various encryption algorithms are elaborated that introduces additional security within DSS. Based on them, security model is constructed that predicts all vulnerable potential points for attack in the system. Hence, classification is done for which security method is adequate to deal with what kind of attack. And last, performance analysis of the DSS system is provided considering all mention coding schemes.

REFERENCES

[1] Weatherspoon, H., Kubiatowicz, J.: Erasure coding vs. replication: A quantitative comparison, In: *Proc. 1st Int. Workshop Peer-to-Peer Syst.* (IPTPS), 2001, pp. 328–338.

[2] Dimakis, A. G., Godfrey, P. B., Wu, Y., Wainright, M. J., Ramchandran, K.: Network coding for distributed storage systems, *IEEE Trans. Inf. Theory*, Vol. **57**, no. 8, pp. 5227–5239 (Aug. 2011).

[3] Sathiamoorthy, M., Asteris, M., Papailiopoulos, D., Dimakis, A. G., Vadali, R., Chen, S., Borthakur. D.: Xoring elephants: Novel erasure codes for big data. In: *Proc. of the VLDB Endowment*, Vol. **6**, pp. 325–336 (2013).

[4] Dimakis, A. G., Prabhakaran, V., Ramchandran, K.: Decentralized erasure code for distributed storage, In: *Trans. Inf. Theory IEEE/ACM Netw*. (June 2006).

[5] Paunkoska, N., Finamore, W., Karamachoski, J., Puncheva, M., Marina, N.: *Improving DSS Efficiency with Shortened MSR Codes*, ICUMT, 2016.

[6] Rashmi, K. V., Shah, N. B., Kumar, P. V.: Optimal exact–regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction, *IEEE Trans. Inf. Theory*, Vol. **57**, no. 8, pp. 5227–5239 (Aug. 2011).

[7] Ylonen, T., Kivinen, T., Saarinen, M., Rinne, T., Lehtinen, S.: SSH Protocol Architecture. Technical report, *The Internet Engineering Task Force IETF*, 2002. http://www.ietf.org/internet-drafts/draft-ietf-secsharchitecture-13.txt.

[8] Dierks, T., Allen, C.: The TLS protocol version 1.0. Technical report, *The Internet Engineering Task Force* IETF, 1999. http://www.ietf.org/rfc/rfc2246.txt.

[9] IPSec Working Group. IP security protocol (IPSec). Technical report, *The Internet Engineering Task Force IETF*, 2002. http://www.ietf.org/html.charters/ipsec-charter.html.

[10] Kher, V., Kim, Y.: Securing distributed storage: challenges, techniques, and systems, *StorageSS '05 Proceedings of the 2005 CM Workshop on Storage Security and Survivability*, 2005, pp. 9–25.

[11] Rawat Ankit Singh: *New coding techniques for distributed storage systems: enabling locality, availability and security*, PhD diss., The University of Texas at Austin, UT Elec. Theses and Diss, 2015.

[12] Seitz, L., Pierson, J., Brunie, L.: Key management for encrypted data storage in distributed systems, *Second IEEE International Security in Storage Workshop*, 2003.

[13] Parra, J. R., Chan, T. H., Ho, S.: Updatable encryption in distributed storage systems using key-homomorphic pseudorandom functions, *Int. J. Infor. and Coding Theory*, Vol. **3**, no. 4, pp. 365–391 (2016).

[14] Kumar, S., Rosnes, E., Amat, A. G.: *Private Information Retrieval in Distributed Storage Systems Using an Arbitrary Linear Code*, ISIT, 2017.

[15] Partala, J.: *Semantically Secure Symmetric Encryption with Error Correction for Distributed Storage, Security and Communication Networks*, Vol. **2017**, Article ID 4321296, 10 pages, 2017.

[16] Dikaliotis, T. K., Dimakis, A. G., Ho, T.: Security in distributed storage systems by communicating a logarithmic number of bits, *2010 IEEE Inter. Sym. on Inf. Theory, Austin, TX*, 2010, pp. 1948–1952.

[17] Pletka, R., Cachin, C.: Cryptographic security for a high-performance distributed file system, *24th IEEE Conference on MSST*, 2007.

[18] Stading, T.: Secure communication in a distributed system using identity based encryption, *CCGrid 2003. 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003. Proc*., 2003.

[19] Shah, N. B., Rashmi, K. V., Kumar, P. V.: Information-theoretically secure regenerating codes for distributed storage, *GLOBECOM 2011*, pp. 1–5.

[20] Goparaju, S., Rouayheb, S. E., Calderbank, R., Poor, H. V.: *Data secrecy in distributed storage systems under exact repair*, (NetCod), Calgary, 2013, pp. 1–6.

[21] Paunkoska, N., Kafedziski, V., Marina, N.: Improving the secrecy of distributed storage systems using interference alignment, In: *IWCMC*, 2018.

[22] Rawat, A. S., Koyluoglu, O. O., Silberstein, N., Vishwanath S.: Optimal Locally Repairable and Secure Codes for Distributed Storage Systems, In: *IEEE Trans. Inf. Theory*, Vol. **60**, no. 1, pp. 212–236 (2014).

[23] Pawar, S., El Rouayheb, S., Ramchandran, K.: Securing dynamic distributed storage systems against eavesdropping and adversarial attacks, In: *IEEE Trans. Inf. Theory*, Vol. **57**, no. 10, pp. 6734–6753 (Oct. 2011).

[24] Pawar, S., El Rouayheb, S., Ramchandran, K.: Securing dynamic distributed storage systems from malicious nodes, *IEEE Intern. Sym. on Infor. Theory Proceedings, St. Petersburg*, 2011, pp. 1452–1456.

[25] Marina, N., Paunkoska, N., Velkoska, A.: Adversarial attacks in the twin-code framework, In: *ICUMT*, 2016.

[26] Kadhe. S., Sprintson, A.: Weakly secure regenerating codes for distributed storage, *Inter. Sym. on Net. Coding* (NetCod), 2014, pp. 1–6.

[27] Paunkoska, N., Kafedziski, V., Marina, N.: Improved perfect secrecy of distributed storage systems using interference alignment, *ICUMT*, 2016.

[28] Paunkoska, N., Marina, N., Finamore, N., Karamachoski, N.: Secure shortened MSR codes, In: *IWCMC*, 2016.

[29] Tandon, R., Amuru, S., Clancy, T. C., Buehrer, R. M.: Toward optimal secure distributed storage systems with exact repair, In: *IEEE Tran. Inf. Theory*, Vol. **62**, No. 6, pp. 3477–3492 (June 2016).