Journal of Information Security and Applications 000 (2017) 1–8

Contents lists available at ScienceDirect

Journal of Information Security and Applications

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

Generalised regenerating codes for securing distributed storage systems against eavesdropping

Jian Xu*, Yewen Cao, Deqiang Wang

School of Information Science and Engineering, Shandong University, Jinan, China

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Distributed storage system
Regenerating code
Generalised regenerating code
Secrecy capacity
Security level

ABSTRACT

Regenerating codes (RCs) are efficient at both storage cost and repair bandwidth, and thus are regarded as preferable candidates for distributed storage systems (DSSs). For DSSs with RCs, a file stored across n distributed nodes can be reconstructed from k ($< n$) nodes. The collection of the k nodes is called the reconstruction set. A failed node can be regenerated (i.e., repaired) from d ($< n$) remaining nodes. The collection of the d nodes is called the regeneration set. In traditional RCs, the numbers of reconstruction sets and regeneration sets are fixed to some specific values. In this paper, we introduce the concept of generalised RCs, in which the value ranges of the numbers of both reconstruction sets and regeneration sets are extended. Compared to traditional RCs, the generalised RCs possess more coding schemes and better system security level in terms of the probability of revealing original data file. An explicit construction of generalised RCs is provided, in which the numbers of both reconstruction sets and regeneration sets can be designed flexibly. Furthermore, based on the generalised RCs, an intruder model where an eavesdropper can access to some nodes is considered and a general upper bound on secrecy capacity is derived. The relationship between the obtained upper bound and existing ones achieved by traditional RCs is discussed in detail. The provided explicit construction is the first optimal construction of generalised RCs, which achieves the upper bound on secrecy capacity and has the flexibility in designing the numbers of reconstruction sets and regeneration sets.

1. Introduction

Distributed storage system (DSS) is an emerging technology relevant to cloud computing and has drawn a lot of attention from the telecommunications industry [1–3]. In DSSs, storage nodes are distributed across a wide geographical area and connected by a network. These storage nodes are individually unreliable (due to many factors, such as disk failures or peer churning in peer-to-peer storage systems [4]) and collectively used to reliably store data files over a long period of time. Examples of these systems include peer-to-peer storage clouds and large data centers [4]. In order to improve the system reliability, data are stored redundantly on the storage nodes. Besides, a self-sustaining DSS must be able to regenerate (i.e., repair) failed nodes. The most straightforward strategy is replication, in which the data are duplicated and stored in multiple storage nodes [5]. This method, though simple, has low storage efficiency. The other strategy is erasure coding which provides better storage efficiency [6]. However, in the face of repairing a failed storage node, erasure coding wastes bandwidth. Both proactive secret sharing [7,8] and regenerating codes (RCs)[4,9] can achieve the similar purpose of security guarantees. However, secret sharing often performs with high redundancy [10]. In contrast, the RCs are efficient at both storage and repair bandwidth (i.e., the amount of data downloaded to repair a failed node) [9,11,12]. Thus, RCs can be preferable candidates for DSSs in terms of data security, storage and repair bandwidth efficiencies.

In traditional RCs, a file stored across n distributed nodes, can be recovered from any k out of n nodes. The collection of k nodes is called the reconstruction set. If a node fails, any d out of the remaining $(n-1)$ nodes can be used to repair the lost data previously stored on the failed node. The collection of d nodes is called the regeneration set. It is generally required that in traditional RCs, the number of reconstruction sets, $\mathcal{N}_{\mathcal{R}}$, and the number of regeneration sets, $\mathcal{N}_{\mathcal{H}}$, satisfy $\mathcal{N}_{\mathcal{R}} = \mathrm{C}_n^k$ and $\mathcal{N}_{\mathcal{H}} = \mathrm{C}_{n-1}^d$. These requirements of reconstruction and regeneration are preferable for legitimate users who want to reconstruct the original data; however these requirements are rigorous for designing coding schemes, resulting in limited available code constructions. On the other hand, these requirements can be adverse to data security in the presence of an intruder that can eavesdrop on some nodes, as any k out of n nodes (i.e., there are C_n^k reconstruction sets) can be used to recover the whole original data file.

* Corresponding author.
E-mail addresses: jianxusdu@126.com (J. Xu), ycao@sdu.edu.cn (Y. Cao), wdq_sdu@sdu.edu.cn (D. Wang).

http://dx.doi.org/10.1016/j.jisa.2017.02.002
2214-2126/

In this paper, we introduce the concept of generalised RCs where $\mathcal{N}_{\mathcal{R}}$ and $\mathcal{N}_{\mathcal{H}}$ satisfy $1 < \mathcal{N}_{\mathcal{R}} \leq C_n^k$ and $1 \leq \mathcal{N}_{\mathcal{H}} \leq C_{n-1}^d$, respectively. Obviously, traditional RC is a special case of the generalised RC. Compared to traditional RCs, the proposed generalised RCs have relaxed requirements for data reconstruction and regeneration, thus resulting in more coding schemes. On the other hand, the generalised RCs possess a better security level (SL) and are preferable for securing DSSs against eavesdropping. Here the probability of revealing original data file in the presence of the intruder is referred to system SL. In generalised RCs, only when the eavesdropper (Eve) accessed to specific k nodes (i.e., the k nodes in one reconstruction set), can it obtain the whole original data file. When accessing to other k nodes which are not in one reconstruction set, the original data file cannot be recovered and thus will not leak to the Eve, resulting in a better SL. In practice, different DSS applications have different requirements of SL, such as government's cloud platforms, the most security-sensitive enterprise cloud platforms and other private or public clouds. By designing different values of $\mathcal{N}_{\mathcal{R}}$, the generalised RCs can possess different system SLs. In [13] and [14], failed node regeneration can be realised by accessing to some specific node sets in order to take full advantage of the different storage capacities or different repair bandwidths. While, the number of nodes used for failed node regeneration is not restricted to d, so long as the total data downloaded exceeds a certain threshold. The performances of generalised RCs, such as data security, storage capacity and repair bandwidth, have not been studied by far.

Based on the generalised RCs, we focus on the data security and study a type of intruder model. The results obtained in the present paper are as follows. (a) An explicit construction of generalised RCs is provided, in which $\mathcal{N}_{\mathcal{R}}$ and $\mathcal{N}_{\mathcal{H}}$ can be designed as different values; (b) The general upper bound on secrecy capacity of the intruder model is derived with an information-theoretic proof; (c) With the aim of verifying the generality of the obtained upper bound, the obtained upper bound is specified into traditional RCs case, then is shown with its compactness (i.e., tightness) and the consistency at two special cases with previous upper bounds obtained in traditional RCs. The provided explicit construction of generalised RCs achieving the upper bound on secrecy capacity is optimal in certain parameter regimes. Besides, an example implementation is provided also.

2. Related work

Considerable studies on RCs can be found in the literature [9,12,16–18]. An optimal tradeoff between storage and repair bandwidth was given in the original work of Dimakis et al. [19]. Two special points on this tradeoff curve are termed as minimum storage regenerating (MSR) and minimum bandwidth regenerating (MBR) points, which correspond to the best storage efficiency and the minimum repair bandwidth, respectively. The RCs achieving MSR and MBR points are called MSR and MBR codes, which were constructed in [12]. It has been shown in [20] that the interior points between MSR and MBR points on this tradeoff cannot be achieved under exact repair with optimality.

The RCs permit a failed node to repair by downloading β units of information from each of d storage nodes. To demonstrate, an example inspired by Pawar et al. [4] is provided by Fig. 1, where a maximal distance separable (MDS) code is used to store a file F of 4 symbols $(x_1, \ldots, x_4)$ over a finite field $\mathbb{F}_q$ of size $q = 5$. The message symbols $(x_1, \ldots, x_4)$ are coded by a (4,2)MDS code, and then stored on four different nodes (i.e., nodes 1, 2, 3, 4), with each node having a storage capacity of $\alpha = 2$ symbols. Here we combine input node In_i and output node Out_i into a single vertex, representing storage node i, $i \in [1, 4]$. A data collector (DC) can reconstruct the whole file F by connecting to any $k = 2$ out of $n = 4$ storage nodes. When a storage node fails, the system needs to be repaired by replacing the failed node by a new node (called replacement node). As depicted in Fig. 1, node 2 failed and the stored tow symbols x_3, x_4 were lost. The failed node 2 is then replaced by replacement node $r(2)$. By making node $r(2)$ connect to $d = 3$ nodes (helper nodes), and download $\beta = 1$ symbol from each, the lost data x_3, x_4 can be recovered.

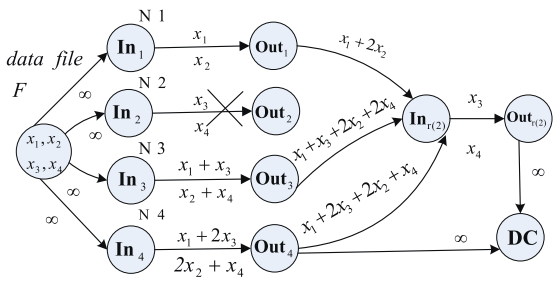

Fig. 1. An example of a DSS under repair. A file F of 4 symbols $(x_1, \ldots, x_4) \in \mathbb{F}_q$ is coded by a (4,2)MDS code and then stored on four different nodes. Denote the four storage nodes as "N i", $i \in [1, 4]$. Combine input node In_i and output node Out_i into a single vertex, representing storage node i. Storage node 2 failed and the stored tow symbols x_3, x_4 were lost. The failed node 2 is then replaced by replacement node $r(2)$, which downloads $(x_1 + 2x_2)$, $(x_1 + x_3 + 2x_2 + 2x_4)$, $(x_1 + 2x_3 + 2x_2 + x_4)$ from nodes 1, 3, 4 respectively, to compute and store symbols x_3, x_4. A data collector (DC) can recover the whole file F by connecting to any $k = 2$ out of $n = 4$ storage nodes. The edges in the graph are labeled by their capacities.

Securing DSSs against an Eve who may come at different time instances during the lifetime of the storage system to observe, has been studied extensively across the literature, such as [4,9,15,16,21]. The aim of security against eavesdropping is to prevent an Eve from obtaining any information about the stored data. The intruder model $\{e_1, e_2\}$DSS where an Eve who can obtain all the downloaded data passed on to a set $\mathcal{E}_1$ of e_1 nodes in the system as these nodes undergo repair, and may also gain read-access to the content of a disjoint second set $\mathcal{E}_2$ of e_2 nodes, is studied in the present paper. Related work on $\{e_1, e_2\}$DSS can be found in traditional RCs case in literature [4,9,21]. The authors in [4] considered an $\{e_1, e_2 = 0\}$DSS, and derived an upper bound on the secrecy capacity of the system. The authors in [9] considered the $\{e_1, e_2\}$DSS and constructed secure MBR and secure MSR codes. The constructed secure MBR codes are shown to be optimal in terms of the upper bound [4]. The authors in [21] derived an upper bound of $\{e_1, e_2\}$DSS at the special case of MSR point, so that the constructed secure MSR codes in [9] were optimal in terms of this upper bound.

The rest of this paper is organized as follows. The framework of generalised RCs and explicit construction of generalised RCs with multi-reconstruction sets and multi-regeneration sets are provided in Section 3. The system model analysis based on information flow graph is presented in Section 4. In Section 5, a general upper bound on secrecy capacity of intruder model in the generalised RCs is derived. The generality of the obtained upper bound is verified under traditional RCs case at MSR and MBR points. In Section 6, an example implementation and the analysis of generalised RCs are given. We conclude the paper in Section 7.

3. Generalised RCs

3.1. Framework of generalised RCs

In this subsection, we firstly introduce the framework of generalised RCs for DSSs. We focus on the repair of single node failure throughout this paper because it is the dominant failure case in DSS [22]. Data are stored across n storage nodes in a distributed manner. Each node has a capacity of α symbols, which is used to

Please cite this article as: J. Xu et al., Generalised regenerating codes for securing distributed storage systems against eavesdropping, Journal of Information Security and Applications (2017), http://dx.doi.org/10.1016/j.jisa.2017.02.002