國立交通大學

資訊科學與工程研究所

碩士論文

分散式網路編碼儲存系統預防線路竊聽問題之研究

A Link Eavesdropping Prevention Problem in Distributed Network Coded Data Storage Systems

研 究 生：廖振宏

指導教授：王蒞君 教授

共同指導教授：王國禎 教授

中華民國 102 年 7 月

分散式網路編碼儲存系統預防線路竊聽問題之研究

A Link Eavesdropping Prevention Problem in Distributed Network Coded Data Storage Systems

研 究 生：廖振宏　　Student：Chen-Hung Liao

指導教授：王蒞君　　Advisor：Li-Chun Wang

共同指導教授：王國禎　　Co-Advisor：Kuo-Chen Wang

國立交通大學
資訊科學與工程研究所
碩士論文

A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

July 2013

Hsinchu, Taiwan, Republic of China

中華民國 102 年 7 月

分散式網路編碼儲存系統預防線路竊聽問題之研究

學生：廖振宏

指導教授：王蒞君
共同指導教授：王國禎

國立交通大學

資訊學院資訊科學與工程研究所

摘要

近年來，雲端運算 (Cloud Computing) 的發展相當的快速，它提供了更多方便且可擴張的服務，雲端分散式儲存系統就是其中之一。在雲端分散式儲存系統中，網路編碼 (Network Coding) 技術扮演著關鍵的角色，它具有高可靠度以及低儲存花費的優點。然而因為其需要更多的遠端修復頻寬 (Remote Repair Bandwidth)，當遠端備份資料中心進行修復時面臨嚴重的線路竊聽問題。在本篇論文中，針對線路竊聽問題，提出最佳化技術的分析模組，依使用者不同安全性需求，得到最小資料儲存量的理論值。我們的結果顯示，使用者安全性需求與儲存花費存在相互影響的理論關係。在此分析模組下，我們進一步探討使用者安全性需求與其他重要儲存系統參數的設計問題。

A Link Eavesdropping Prevention Problem in Distributed Network Coded Data Storage Systems

A THESIS Presented to

The Academic Faculty By

Chen-Hung Liao

In Partial Fulfillment

of the Requirements for the Degree of

Master in Computer Science

Institute of Computer Science and Engineering

College of Computer Science

National Chiao-Tung University

2013

Abstract

In recent years, network coding plays a key role in distributed storage systems, because of high reliability, security, and low storage cost. However, network coding-based distributed storage systems face an eavesdropping problem when transmitting the repairing data from remote datacenters. This problem is especially crucial in distributed network coded storage systems because more repair bandwidth and repair links are required, compared to conventional replication. In this thesis, we propose an optimization approach to compute the minimum storage according to the required security level. Our numerical results demonstrate that there exists an optimal tradeoff between remote repair bandwidth and storage cost. Moreover, we analyze the relation between security level requirement and the number of remote and local storage nodes, storage cost, data reliability, and secrecy capacity.

Acknowledgments

I would like to thank my parents and younger brother. They always give me endless supports and warm encouragement. I especially thank Professor Li-Chun Wang who gave me many valuable suggestions in my research during these two years. I would not finish this work without his guidance and comments. Professor Li-Chun Wang not only gave me suggestions in research, but also many experience about life. I really learned a lot.

I am deeply grateful to my laboratory mates, Yin-Ming, Shao-Heng, Cheng-Wen, Yi-Tsen, Yu-Chia, Chia-Yu, Gen-Hen ,and junior laboratory mates at Mobile Communication and Cloud Computing Laboratory at the Graduate Institute of Communications Engineering and the Graduate Institute of Computer Science and Engineering in National Chiao-Tung University. They provide me much assistance and share much happiness with me.

Contents

List of Tables

List of Figures

Glossary of Symbols

- n: number of total storage nodes in the storage system.
- k: coding parameter using (n, k) code in the storage system.
- Ω: original data object size.
- d: number of the surviving nodes in the storage system.
- N_L: number of storage nodes used to store the data in local datacenter.
- N_R: number of storage nodes used to store the data in remote datacenter.
- $\gamma_R(i)$: remote repair bandwidth for the i-th new-comer.
- γ_R: constant value of remote repair bandwidth.
- $\gamma(i)$: total repair bandwidth for the i-th new-comer.
- γ_R: constant value of remote repair bandwidth.
- β_L: amount of downloaded bits from local datacenters.
- β_R: amount of downloaded bits from remote datacenters.
- m: ratio of capacity of local/remote link.
- $M_L(i)$: number of links from local datacenters for the i-th new-comer.
- $M_R(i)$: number of links from remote datacenters for the i-th new-comer.
- M_L: constant value of the number of links from local datacenters.

- M_R: constant value of the number of links from remote datacenters.
- λ: security parameter.
- σ: security level requirement.
- α: storage per node.
- α^*: minimum storage per node according to remote repair bandwidth.
- $\widetilde{\alpha^*}$: minimum storage per node under required security level.
- $\widetilde{\gamma_R}$: maximum remote repair bandwidth according to data object size and security level requirement.
- $\gamma_{R,\min}$: minimum remote repair bandwidth.
- α_{MBR}: storage per node at minimum remote repair bandwidth point.
- α_{MSR}: storage per node at minimum storage point.
- $\gamma_{R,MBR}$: remote repair bandwidth at minimum remote repair bandwidth point.
- $\gamma_{R,MSR}$: remote repair bandwidth at minimum storage point.

CHAPTER 1

Introduction

1.1 Motivation

With the flexibility of allocating computing and communications resources, cloud computing is changing the paradigm of the future development of information and communication technologies. Cloud computing provides on-demand measured services by location independent resource pooling. Cloud storage services are a popular and important cloud application, such as Dropbox and Google Drive. The benefits of moving data into cloud servers include relieving the burden of storage resources, global data access, and avoiding huge expenditure on the infrastructure [1]. A global storage infrastructure, OceanStore can automatically recovers from server and network failures [2]. Users do not have to carry huge amount of data around. They can just access in the cloud, instead.

In a cloud distributed storage system, data are distributed to multiple storage nodes interconnected in a network [3], [4], [5]. Important issues for cloud storage system are data reliability and security [2], [6]. A common approach for enhancing data reliability is to distribute data across multiple datacenters and introduce redundancy to tolerate possible failures. Furthermore, the mechanism of repairing failures, *repair process*, is essential when a storage node does not function well. A new storage node, *new-comer*, downloads data from other surviving storage nodes, and regenerate data to replace the failed node. During a repair process, the number of bits that a new-comer downloads from surviving storage nodes is called *repair bandwidth*.

In the literature, different strategies to provide data reliability have been proposed, like replication and erasure coding [7]. Replication is the simplest and most common way for redundancy, which just replicates data with multiple storage nodes. However, erasure coding techniques are shown to achieve higher reliability than replication for the same redundancy [8], [9]. In recent years, network coding techniques, which combine data in intermediate nodes, were proposed to reduce repair bandwidth compared to standard erasure codes [10]. Dimakis et al. further derived a tradeoff between storage and repair bandwidth and showed that network codes can achieve the optimal tradeoff curve [11], [12]. Q. Yu et al. analyzed the tradeoff curve based on Dimakis et al.'s work [13].

However, network coding produces more repair bandwidth than simply replication. Also, a new-comer has to connect to more nodes to download data fragments than conventional erasure coding does. Recent studies (e.g., [14], [15], [16]) considered the storage node eavesdropping (malicious node) problem. That is, the storage nodes will be invaded by an eavesdropper or compromised by an adversary during a repair process. If an eavesdropper observes a node that is added to the system to replace a failed node, it will have access to all the data downloaded during repair, which can potentially compromise the entire information in the system.

1.2 Problem and Solution

In this thesis, instead of focusing on node eavesdropping problem, we address the link eavesdropping problem for cloud inter-datacenter distributed storage systems. *Inter-*

datacenter scenario represents that data are stored in multiple different datacenters in different regions for increasing reliability as shown in Figure 1.1. By doing this, cloud storage systems can guarantee data accessible and recoverable even if a disaster happens to local datacenter. This methodology is called *remote backup*. The local datacenter plays an role of cache server for main service and the remote data center is used for remote backup.

There exists a security problem during a repair process in such scenario. When a storage node fails in a local datacenter, a new-comer downloads data fragments from local and remote datacenters in different regions and generates new data fragments to replace the failed node. The number of bits that a new-comer downloads from surviving storage nodes in remote datacenters during a repair process is called *remote repair bandwidth*. Since the repairing data of remote repair bandwidth are transmitted over the Internet, the communication between the local and remote datacenter can become susceptible to eavesdropping. An eavesdropper can exactly know the original data as long as he/she collects enough network coded data [17]. This thesis focuses on such scenarios where an eavesdropper can gain complete information of remote repair bandwidth. Under this setting, the remote repair bandwidth is a major factor affecting the system security level.

This problem is crucial in the network coded distributed storage systems because more repair bandwidth and repair links are required during the repair process. How can we evaluate and reduce the risk of leaking data to eavesdroppers in this case. Is it possible not to reveal any information to eavesdroppers so that the system can achieve perfect secrecy. In this thesis, we show that remote repair bandwidth can be reduced by increasing storage per node and derive the tradeoff curves between remote repair bandwidth and storage. The minimum storage for achieving required security level can be also given. We further show analysis of the relation between security level requirement and important system parameters such as the number of remote and local storage nodes, storage cost, data reliability, and secrecy capacity.

1.3 Thesis Outline

The rest of this thesis is organized as follows. In Chapter 2, we describe the background of redundancy techniques, network coding, and discuss related work on node eavesdropping problem. Chapter 3 introduces our system model and problem formulation. In Chapter 4, we give the storage optimization analysis and the relation between security requirement and some important system parameters. In Chapter 5, we show and discuss the numerical results for the relation between security requirement and some important system parameters. We conclude the thesis and provide some suggestions for future research in Chapter 6.

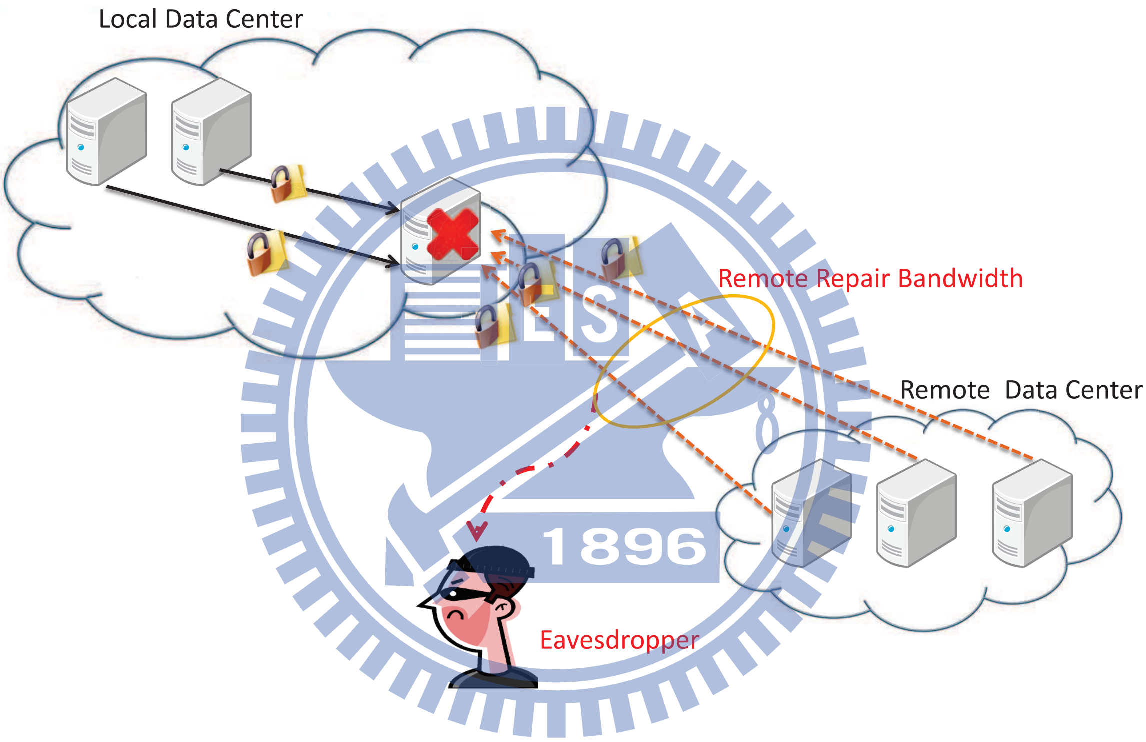

Figure 1.1: The eavesdropping problem for data repairing in inter-datacenter.

CHAPTER 2

Background

2.1 Replication

Replication is the simplest and most common way for redundancy in reliable storage systems. When a user stores a data object in a distributed storage system based on replication, the system replicates the source data object into r replicas (r is called replicate ratio) and then these replicas are distributed to the storage nodes. Every storage node stores an entire copy of the source data object. This method, though simple, has huge storage cost. It needs r times storage space to store single data object [9].

2.2 Erasure Coding

Erasure coding is another usual way to generate redundancy. It does not just replicate the data object. In contrast, it first divides the data object into k fragments, and then encodes them into n encoded fragments. Finally, these encoded fragments are distributed to the n storage nodes, where $n > k$. Any legal user can access any k out of these encoded fragments, and reconstruct the original data object via some computation [8]. Erasure

Table 2.1: Comparison between replication and erasure coding under the same fault tolerant ability.

	Replication (replicate 4)	Erasure Coding(7,4)
Fault Tolerant Ability	$r-1$ blocks (3)	$n-k$ blocks (3)
Storage Space	$k*r$ blocks (16)	n blocks (7)
Repair Bandwidth	1 block (1)	k blocks (4)

coding provides higher reliability and costs less storage space than replication. However, erasure coding produces higher repair bandwidth than replication. Table 2.1 shows the comparison between replication and erasure coding under the same fault tolerant ability. We select replicate ratio 4 and (7,4) erasure code to illustrate the comparison.

Here we give an example to illustrate the repair process using only (4,2) erasure code (see Fig. 2.1). Consider a storage system which contains four storage nodes. Assume the size of the data object is 4 MB. Upon the data object is to be stored, it first will be divided into four fragments in equal size, and then encoded into eight fragments. Note that a legal user can collect any four out of these eight fragments to reconstruct the original data object. Second, these fragments will be stored in four storage nodes in distributed way. During the repair process, a new-comer can connect to any two storage nodes to download four fragments to reconstruct the original data object. Since each storage node stores two fragments, the storage per node is 2 MB. The new-comer totally downloads four fragments, so the repair bandwidth is 4 MB, which is equal to the original data object size [18].

2.3 Network Coding

Network coding is a generalization of the conventional routing (store-and-forwarding) method [11]. In conventional routing, each intermediate node in the network simply stores and forwards the information received. In contrast, network coding allows the intermediate nodes to generate output data by encoding previously received data. An intermediate node can function as an encoder in the sense that it receives information from all the input links, encodes, and sends information to all the output links [19]. Thus, network coding allows information to be mixed at intermediate nodes. We refer to coding at a node in a network as network coding. Network coding can be used to improve the network robustness [20], [21], [22], network throughput [21], and confidentiality [23].

In recent years, the concept of combining network coding with distributed storage while downloading data fragments has been introduced [11]. Such coding scheme is called *Regenerating Code*. In this scheme, the repair bandwidth can be reduced rapidly [18]. Figure 2.2 gives another example using network coding. The original data object is encoded and stored as conventional erasure coding. The difference is that the data fragments to be downloaded are put in packets, and the packets in same storage node are mixed before transmitted to a new-comer.

In Fig. 2.2, a new-comer connects to three storage nodes. The data fragments are mixed and then three packets are transmitted to the new-comer. The storage per node is the same as the example in Fig. 2.1, but the repair bandwidth is reduced to 3 MB. That

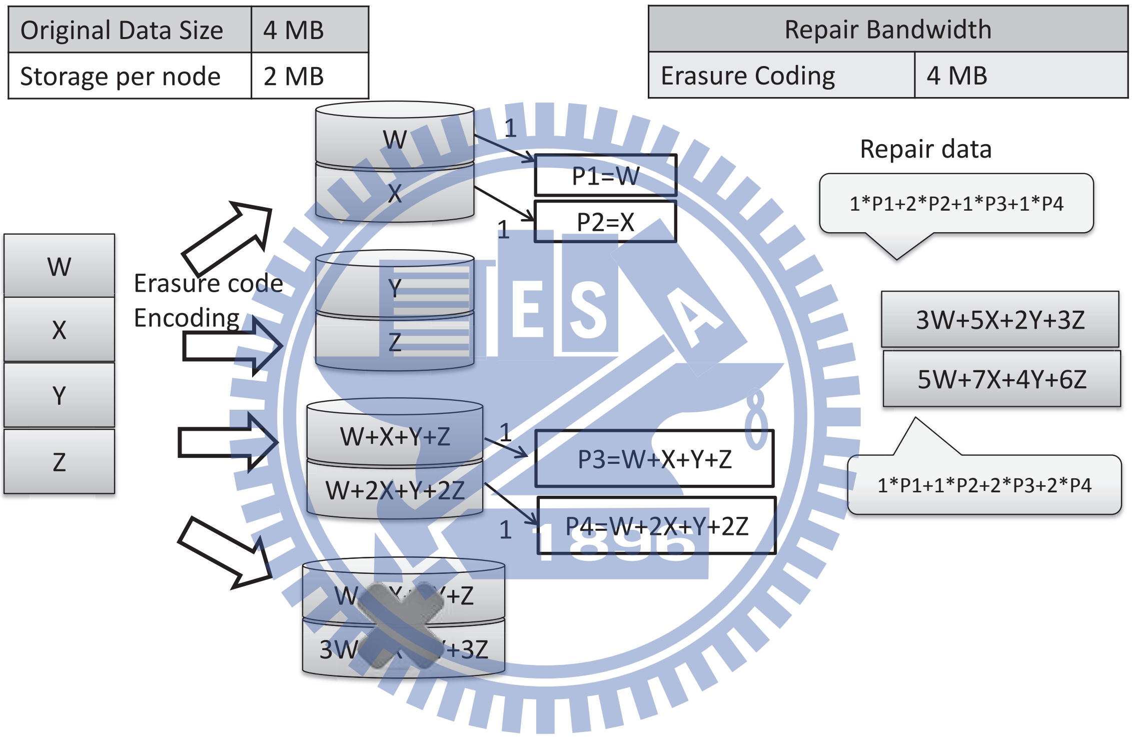

Figure 2.1: Repair bandwidth using only erasure coding.

is, using network coding reduces 25% of the repair bandwidth. Previous work further identified there is a fundamental tradeoff between storage and repair bandwidth [11]. However, as we have shown in the example, network coding still causes more repair bandwidth than using simply replication, and a new-comer has to connect to more nodes to download data fragments than conventional erasure coding.

2.4 Literature Survey

Storage nodes in a distributed storage system may not be secure and may be susceptible to an intruder that can eavesdrop on the nodes and possibly modify their data. The intruder can observe a node that is added to the system to replace the failed node and can access to all the data downloaded during repair, which can potentially compromise the entire information in the system. T.K. Dikaliotis et al. and K. Rashmi et al. indicate the problem of maintaining an encoded distributed storage system when some nodes contain errors or erasures, and provide maximum detectable, tolerable errors and erasures [24], [25]. Y. Wu et al. present techniques for constructing codes that achieve the optimal tradeoffs between storage efficiency and repair bandwidth [26]. S. Jaggi et al. indicate the problem that if the network scheme with network coding contains hidden malicious nodes that can eavesdrop on transmissions and inject fake information, it will cause a decoding error [27], [28]. S. Pawar et al. determine the secrecy capacity (i.e., the maximum amount of data that can be securely stored and made available to a legitimate user without revealing any information to any eavesdropper) of distributed storage systems under repair dynamics [14]. N.B. Shah et al. provide an explicit product-matrix code constructions that achieve information-theoretic secrecy capacity [15]. T. Ernvall et al. study the secrecy capacity of heterogeneous distributed storage systems (i.e., nodes have different storage capacities and different repair bandwidths) in which nodes may be compromised by an eavesdropper [29]. The upper-bounds of the maximum amount of information that can be

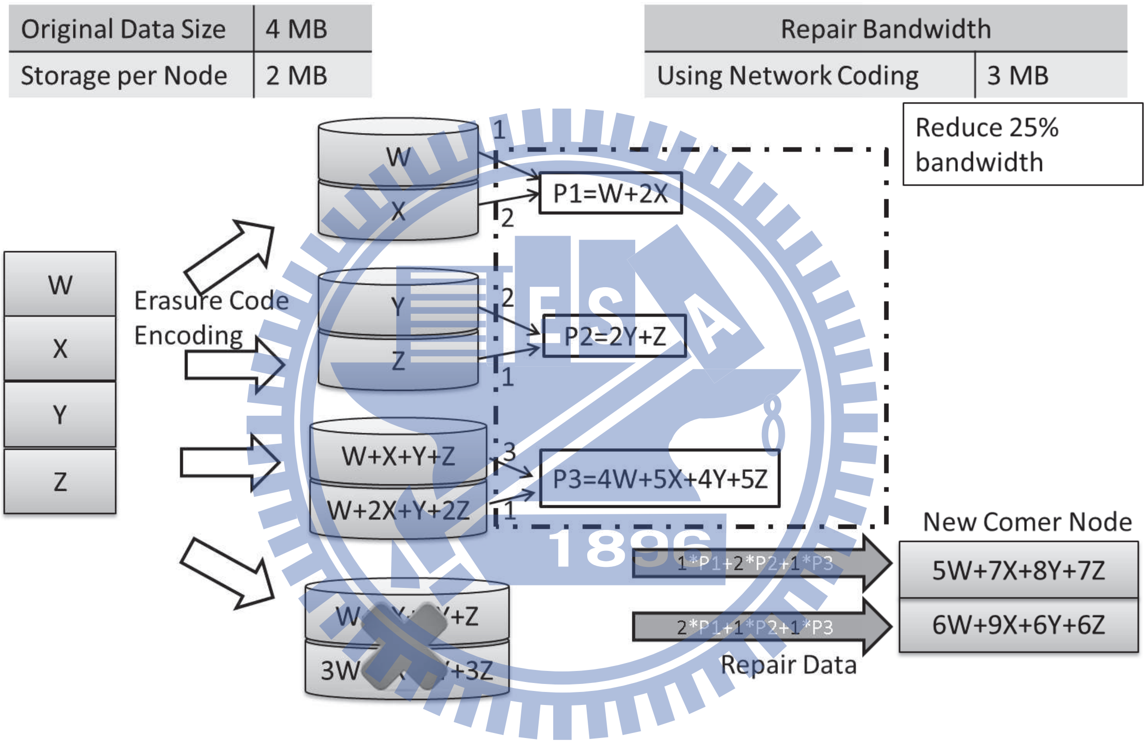

Figure 2.2: Repair bandwidth using network coding.

stored safely on a distributed storage systems against a passive eavesdropper observing a fixed number of nodes is given in [14], [30].

However, all previous works focus on node eavesdropping in the same region cloud datacenter. That is, eavesdroppers can invade cloud datacenters and observe data downloaded by the new-comer node or data stored in the surviving nodes. In this thesis, we consider inter-datacenter scenario where data are distributed in different datacenters in different regions and identify a link eavesdropping problem when repairing data are transmitted over the untrusted wide area network. This problem is crucial in the network coded distributed storage systems because more repair bandwidth and repair links are required.

CHAPTER 3

System Model and Problem Formulation

3.1 System Model

3.1.1 System Scenario

We now introduce the system scenario and notations used in this thesis. The considered inter-datacenter scenario consists of a local datacenter and a remote datacenter. We assume that there exists total of n storage nodes in the two datacenters with N_L storage nodes and N_R storage nodes in the local datacenter and the remote datacenter, respectively.

The system scenario is stated as follows. A user uploads a data object of size Ω to the datacenters. The data is encoded by using (n, k) code and then distributed to the storage nodes in local and remote datacenters. Each storage node stores encoded data fragments of size α. We assume there are some storage nodes failed in local datacenter, and d storage nodes still survive in total. For maintaining the same level of reliability, the system creates new-comer nodes to replace the failed nodes. The i-th new-comer node connects to the surviving storage nodes with $M_L(i)$ nodes from local datacenter and $M_R(i)$ nodes from remote datacenter, so $M_L(i) + M_R(i) = d$. A new-comer downloads β_L bits from each node in local datacenter, and β_R bits each node in remote datacenter. Without loss of generality, we let $\beta_L = m\beta_R$, where $m \geq 1$, considering the local link's capacity is larger than remote link's capacity. Furthermore, we define remote repair bandwidth for i-th new-comer as $\gamma_R(i)$ which equals to $M_R(i)\beta_R$ explicitly.

Finally, we define security parameter as λ, which is the probability that user's data can be reconstructed by an eavesdropper during repair process, and user-specified security level requirement as σ, which is the security rate that storage system can prevent an eavesdropper from reconstructing original data when he/she can observe remote repair bandwidth. We will introduce both security parameter and security level requirement in section 4.3.

3.1.2 Information Flow Graph

Now we are ready to model the link eavesdropping problem. We first introduce the information flow graph for the considered inter-datacenter scenario. In the later section, we will derive the tradeoff curve by solving an optimization problem subject to a sufficient flow constraint. Finally, we give the minimum storage per node for achieving required security level.

We model the inter-datacenter scenario as an *information flow graph*. Our model is based on this particular graphical representation of a distributed inter-datacenter storage system. The information flow graph describes how the information of the data object is communicated through the storage network, stored in nodes with limited storage, and reaches reconstruction points at the data collectors.

The information flow graph is a directed acyclic graph consisting of three kinds of nodes: a single data source node S, storage nodes component x_{in}^i and x_{out}^i, and data collectors DC_i. The i-th storage node in the system is represented by a storage input node x_{in}^i and a storage output node x_{out}^i in the graph. These two node are connected by a directed edge (x_{in}^i, x_{out}^i) with capacity equal to the amount of data stored at the i-th storage node. The capacity of each storage node is α.

When a storage node failure occurs, a repair process is initiated to repair the failure node. This consequently causes the flow graph to be dynamic and evolving with time. At any given time, the activity of a node in the information flow graph depends on whether the node is failed or not. In the initial state, only the source node S is active and it chooses an initial set of storage nodes which connects to their input nodes (x_{in}^i) with outgoing directed edges of infinite capacity. From this point afterwards, the original source node becomes inactive and the initial chosen storage nodes become active. When the i-th storage node fails in the system, the corresponding nodes component x_{in}^i and x_{out}^i become inactive in the graph. New-comer nodes join the system and connect to active nodes. The components of the j-th new-comer node are represented as x_{in}^j and x_{out}^j with the edge $\left(x_{in}^j, x_{out}^j\right)$ added to the information flow graph. Figure 3.1 shows the information flow graph with new-comer. The new-comer chooses to connect with d active nodes to download the encoded data. If the j-th new-comer node chooses to connect to the i-th storage node, we add a directed edge from x_{in}^i to x_{in}^j with capacity equal to the amount of information communicated from node i to the new-comer. We denote the capacity of this edge as β_L and β_R if the new-comer connects to a storage node from the local datacenter and the remote datacenter, respectively. A data collector (DC) is represented by a node connected to k active storage output nodes through infinite capacity links enabling it to download all their stored data and reconstruct the original data object.

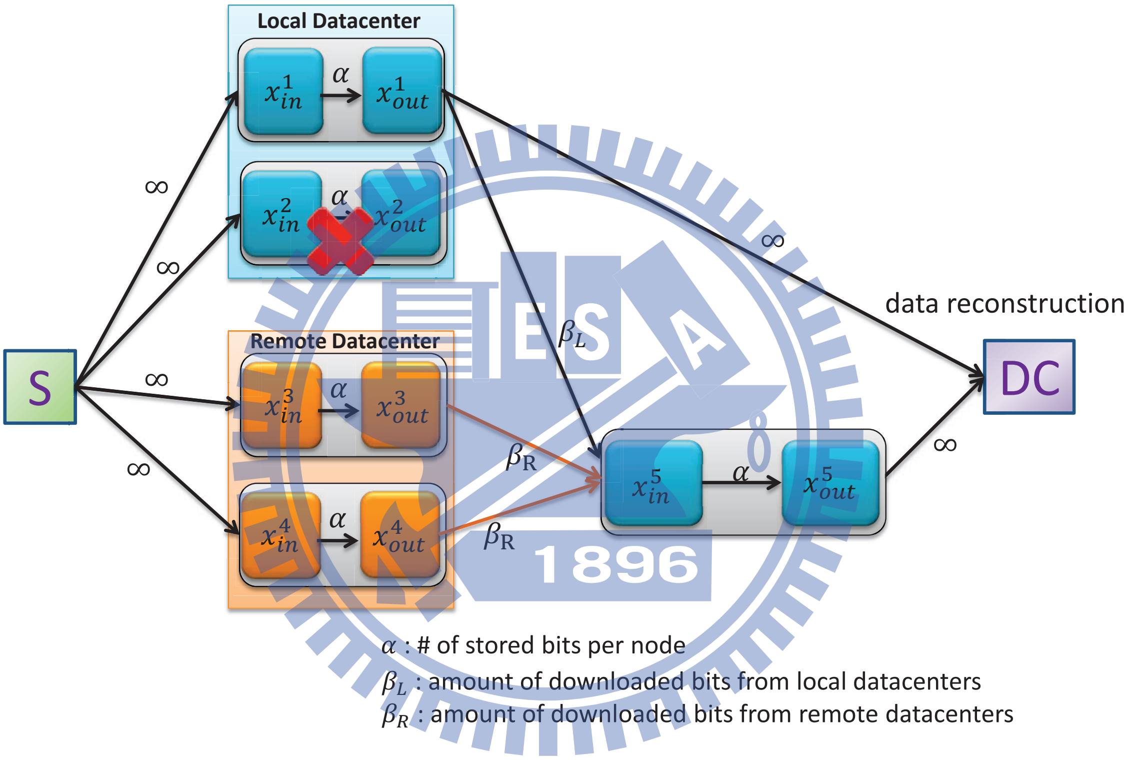

Figure 3.1: Information flow graph for remote repair.

3.2 Problem Formulation

In an inter-datacenter distributed storage system, when a storage node fails, the repair process is executed. The new-comer node gathers data fragments from local and remote datacenters to replace the failure node. Downloading data fragments from remote datacenters may be risky because of eavesdropping, so remote repair bandwidth is an important factor of leaking privacy. Remote repair bandwidth can affect the amount of information that eavesdroppers can obtain. Eavesdroppers can reveal original data object by collecting sufficient information from remote repair bandwidth. The more information they obtain, the higher probability they can decode the original data object.

Therefore, the storage system security level is related to remote repair bandwidth. Our objective is to analyze the minimum storage per node under a required security level. In this thesis, we make four assumptions to discuss our problem:

- While a user uploads his/her data to the storage system, the system coding method (n, k) is decided. After the system finishes storing encoded data, the value of N_L and N_R are also decided. Note that we always assume that the eavesdropper has a complete knowledge of the code and the repair scheme implemented in the system.

- We assume a node fails in local datacenter, so the storage system will execute repair process to add a new-comer node to replace the failed node. Then the number of surviving nodes d can be decided.

- Because the storage system can cache the frequently used data or use proxy server to maintain the data temporarily, data usually tend to be stored more in local datacenters. We consider the case that the number of storage nodes in local datacenter is larger than or equal to k, that is, $N_L \geq k$.

- There may be different number of storage nodes in local and remote datacenter. We consider the worst case that a new-comer node downloads data fragments from all the storage nodes in the remote datacenter, which causes the largest remote repair bandwidth $M_R(i)\beta_R$. Therefore, we let the value of each $M_R(i)$ to be a constant value M_R, which equals to N_R. Then $M_L(i)$ can be written as $M_L(i) = d - M_R(i) = d - M_R = d - (n - N_L) = N_L - (n - d)$, which is also defined as a constant value M_L.

CHAPTER 4

Analysis

In this chapter, we use optimization technique to analyze the minimum storage per node under the user-specified security level requirement in inter-datacenter distributed storage system. In the following, we first derive the storage optimization constraint. Second, we solve the optimization problem and find the tradeoff between storage per node and remote repair bandwidth. Finally, we give definition of security parameter and security level requirement, and find the relation between storage per node and security level requirement.

4.1 Storage Optimization Constraint

4.1.1 Minimum Cut in Information Flow Graph

We now introduce the minimum cut of the information flow graph. A cut in the graph between the source S and a fixed data collector node DC is a subset C of edges such that, there is no directed path starting from S to DC that does not have one or more edges in C. The minimum cut is the cut between S and DC in which the total sum of the edge capacities is smallest.

4.1.2 Flow Constraint

Here, we derive the flow constraint for the considered optimization problem. Next, we give the solution steps of the optimization problem. We define flow constraint :

Definition 1 (flow constraint) *A data collector that reconstruct the original data object successfully must satisfy this constraint :*

$$\mathrm{mincut}(S, DC) \geq \Omega \ , \tag{4.1}$$

where Ω is the original data object size. That is, no data collector DC can reconstruct the original data object if the minimum cut in the information flow graph between S and DC is smaller than the original data object size Ω. We know that the information of the original data object must be transmitted from the source to the particular data collector, and every link in the information flow graph can only be used at most once. Since the point-to-point capacity between S and DC is less than the data object size, it can be shown by a standard cut-set bound that the entropy of the data object conditioned on everything observable to the data collector is nonzero. Therefore, it is impossible for the data collector to reconstruct the original data object.

4.1.3 Lower-bound of Minimum Cut

We introduce Lemma 1 (*lower-bound of minimum cut*) to find the lower-bound of the value of the minimum cut in the information flow graph based on the considered scenario.

Lemma 1 *Consider any information flow graph, formed by having initial nodes(including local and remote storage nodes) that connect directly to the source and obtain bits, while* *additional nodes join the graph by connecting to existing nodes and obtaining bits. Any data collector that connects to a k-subset of the output nodes in the graph must satisfy*

$$\text{mincut}(S, DC) \geq \sum_{i=0}^{k-1} \min\left\{\alpha, \left(M_L(i) - i\right)\beta_L + M_R(i)\beta_R\right\} \ . \tag{4.2}$$

We give the proof of Lemma 1 as follows : First, we show that there exists an information flow graph (see Fig. 4.1) where the bound (4.2) is matched with equality. We assume there are initially n nodes labeled from 1 to n in this graph, and then k new-comers labeled as $n+1, ..., n+k$ are added. The new-comer node $n+i+1$ connects to nodes $n+i+1-d, ..., n+i$ and a data collector DC connects to the last k nodes, i.e., nodes $n+1, ..., n+k$. Consider a cut $(E, \overline{E})$ defined as follows. For each $i \in \{0, \ldots, k-1\}$, if $\alpha \leq \left(M_L(i) - i\right)\beta_L + M_R(i)\beta_R$, then we include x_{out}^{n+i+1} in $\overline{E}$. Otherwise, we include x_{out}^{n+i+1} and x_{in}^{n+i+1} in $\overline{E}$. Then we find this cut $(E, \overline{E})$ achieves (4.2) with equality.

Second, we show that (4.2) must be satisfied for any graph G formed by adding d in-degree nodes as described above. Consider a data collector DC that connects to a k-subset of output nodes. We want to show that the capacity of any $S - DC$ cut in G has a lower-bound:

$$\sum_{i=0}^{k-1} \min\left\{\alpha, \left(M_L(i) - i\right)\beta_L + M_R(i)\beta_R\right\} \ . \tag{4.3}$$

Since all the capacities of the incoming edges of DC are infinite, we only need to examine the cuts $(E, \overline{E})$ with $S \in E$ satisfying

$$x_{out}^{i} \in \overline{E}, \forall i \in I \ . \tag{4.4}$$

Let C denote the edges in the cut, i.e., the set of edges going from E to $\overline{E}$. We apply the topological sorting concept in following. There exists a topological sorting in any directed acyclic graph, where a topological sorting (or acyclic ordering) is an ordering of its vertices such that the existence of an edge from v_i to v_j implies $i < j$. Let x_{out}^{1} be the topologically first output node in $\overline{E}$. Consider two cases:

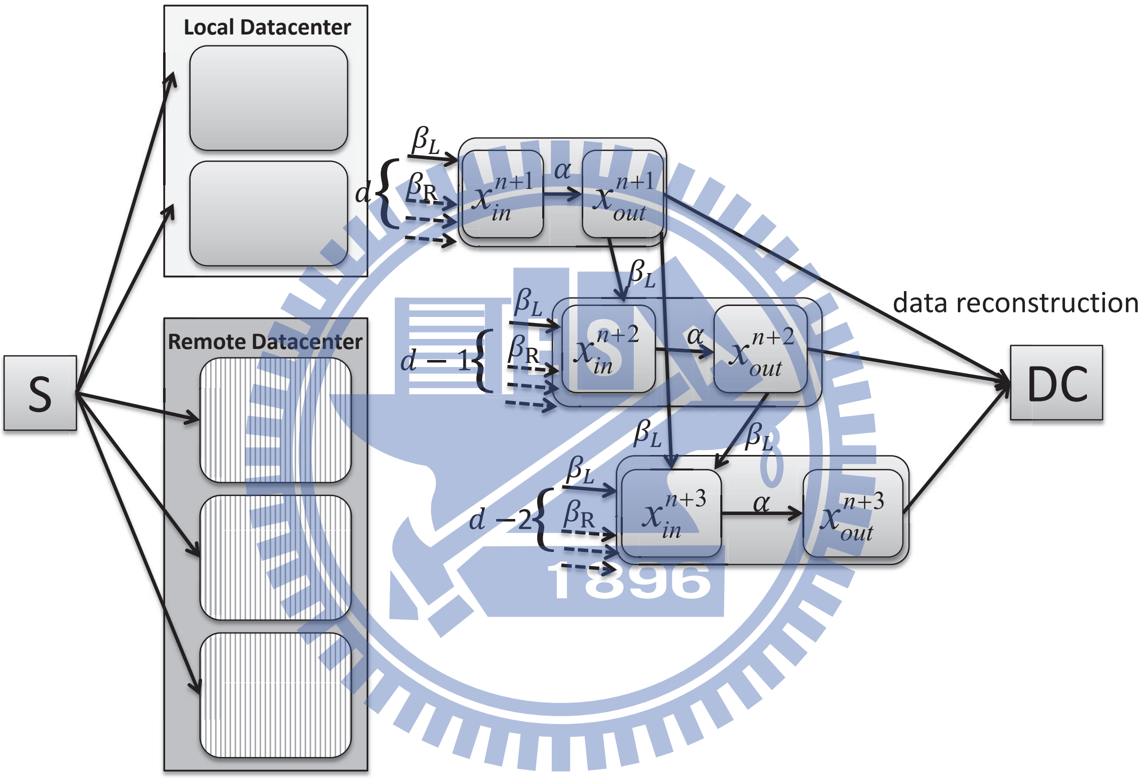

Figure 4.1: Information flow graph for proof of Lemma 1.

1. Case 1 : If $x^1_{in} \in E$, then the edge (x^1_{in}, x^1_{out}) must be in C.

2. Case 2 : If $x^1_{in} \in \overline{E}$, since x^1_{in} has an in-degree of d and it is the topologically fist node in $\overline{U}$, all the incoming edges of x^1_{in} must be in C.

Therefore, these edges related to x^1_{out} will contribute a value of $min\{\alpha, M_L(0)\beta_L + M_R(0)\beta_R\}$ to the cut capacity. Now we consider x^2_{out}, the topologically second output node in $\overline{E}$. Similar to the above, we consider two cases:

1. Case 1 : If $x^2_{in} \in U$, then the edge (x^2_{in}, x^2_{out}) must be in C.

2. Case 2 : If $x^2_{in} \in \overline{U}$, since at most one of the incoming edges of x^2_{in} can be from x^1_{out}, incoming edges of x^1_{in} must be in C.

Therefore, these edges related to x^2_{out} will contribute a value of $min\{\alpha, (M_L(1) - 1)\beta_L + M_R(1)\beta_R\}$ to the cut capacity. Following the same reasoning we conclude that for the i-th node $(i = 0, \ldots .k - 1)$ in the sorted set $\overline{E}$, either one edge of capacity α or $M_L(i) - i$ edges of capacity β_L together with $M_R(i)$ edges of capacity β_R must be in C. Equation (4.3) is exactly summing these contributions. Thus we find the lower-bound of the value of the minimum cut. The illustration is in Figure 4.2.

4.1.4 Storage Optimization Constraint

From the flow constraint and Lemma 1, we can obtain Lemma 2:

Lemma 2

$$\sum_{i=0}^{k-1} \min\left\{\alpha, \left(M_L(i) - i\right)\beta_L + M_R(i)\beta_R\right\} \geq \Omega \quad . \tag{4.5}$$

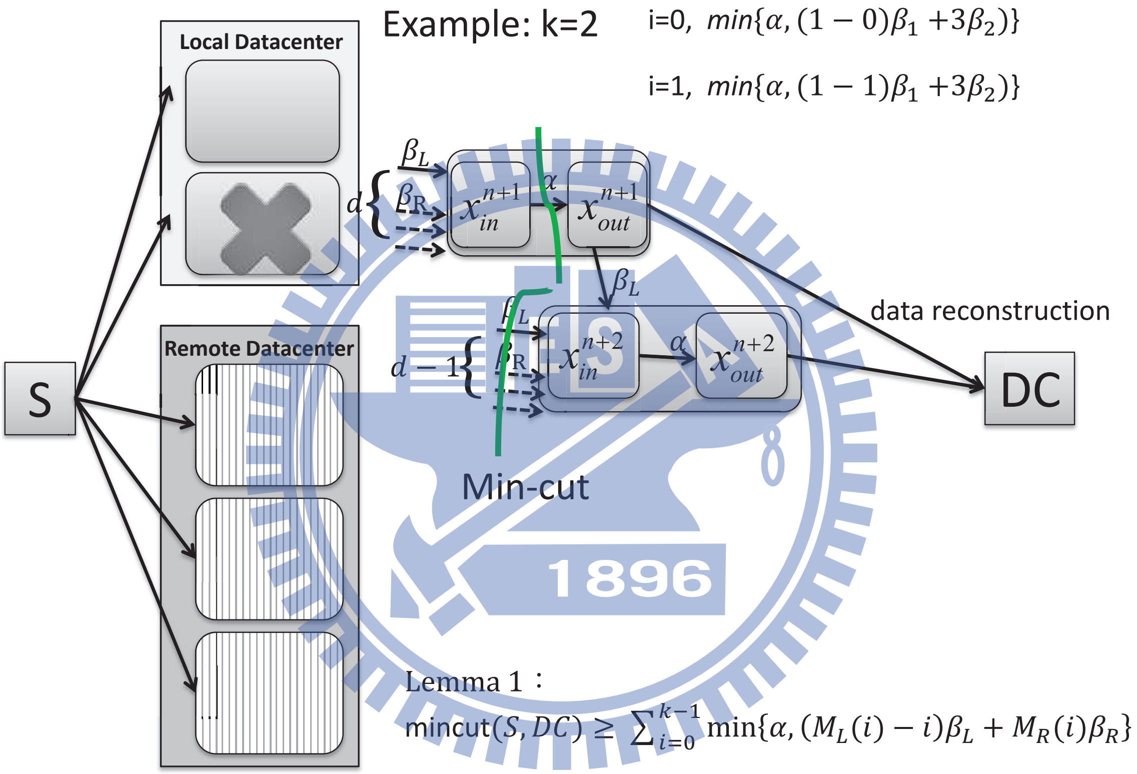

Figure 4.2: Illustration for lower-bound of minimum cut.

We give the proof of Lemma 2 as follows :

Because

$$\sum_{i=0}^{k-1} \min\left\{\alpha, \left(M_L(i) - i\right)\beta_L + M_R(i)\beta_R\right\}$$

is the minimum value of the minimum cut, we can easily know this value must larger than or equal to the original data object size.

In this thesis, we have Lemma 2 as our storage optimization constraint, we will show the solution in the next section.

4.2 Tradeoff Between Storage per Node and Remote Repair Bandwidth

4.2.1 Repair Bandwidth

Based on the information flow graph, we analyze the relation between storage per node and remote repair bandwidth via solving an optimization problem. As described in section 3.2, we made some assumptions before solving the optimization problem. We assume a node is failed in local datacenter, the storage system will execute the repair process to add a new-comer node to replace the failed node. One important observation is that the repair bandwidth can be reduced in network coding based storage system while the new-comer communicate with more storage nodes. While the new-comer communicates with more storage nodes, the size of each communicated packet becomes smaller fast enough to make the repair bandwidth decrease, as d increase, and therefore, minimal for $d = n - 1$. Thus, when the new-comer connects to all surviving nodes, i.e., $d = n - 1$, the repair bandwidth can be reduced most. Most network coded storage systems favor this setting. Also, we consider worst case to make the remote repair bandwidth $M_R(i)\beta_R$ maximize.

So the new-comer should connect to all surviving storage nodes in remote datacenter as possible.

We let $M_R(i)$ for all i to be a constant value M_R which equals to N_R , and then calculate $M_L(i) = d - M_R(i) = d - M_R = d - (n - N_L) = N_L - (n - d)$ which is also a constant value defined as M_L. Furthermore, we let $\beta_L = m\beta_R$, where $m \geq 1$, considering the local link's capacity is larger than remote link's capacity without loss of generality. Then the total repair bandwidth is $\gamma(i) = M_L(i)\beta_L + M_R(i)\beta_R = \beta_R(dm - mN_R + N_R) = \gamma$, and the remote repair bandwidth is $\gamma_R(i) = M_R(i)\beta_R = M_R\beta_R = N_R\beta_R = \beta_R(n - N_L) = \gamma_R$.

4.2.2 Storage Optimization Solution Steps

Here, we try to find the whole region of feasible points (α, γ_R), and then select the one that minimizes storage α. From section (4.1.4), we have Lemma 2 as our storage optimization constraint :

$$\sum_{i=0}^{k-1} \min\left\{\alpha, (M_L(i) - i)\,\beta_L + M_R(i)\beta_R\right\} \geq \Omega\ .$$

Our storage optimization constraint (4.5) can be explicitly solved as follows:

$$\sum_{i=0}^{k-1} \min\left\{\alpha, (M_L(i) - i)\,\beta_L + M_R(i)\beta_R\right\} \geq \Omega$$

$$\Rightarrow \sum_{i=0}^{k-1} \min\left\{\alpha, m\beta_R(d - M_R(i) - i) + M_R(i)\beta_R\right\} \geq \Omega$$

$$\Rightarrow \sum_{i=0}^{k-1} \min\left\{\alpha, \beta_R(md - mM_R(i) - mi + M_R(i))\right\} \geq \Omega$$

$$\Rightarrow \sum_{i=0}^{k-1} \min\left\{\alpha, (\frac{md - mi}{M_R} - (m-1))\gamma_R\right\} \geq \Omega\ ,$$

where

$$\gamma_R = M_R(i)\beta_R = M_R\beta_R = N_R\beta_R \ .$$

We simplify notation in order to make it easier to show detailed steps. We let

$$b_i = (\frac{md - mk + mi + m}{M_R} - (m-1))\gamma_R, \text{for } i = 0, 1, 2, ..., k-1 \ . \tag{4.6}$$

Then the problem is to minimize α subject to the constraint

$$\sum_{i=0}^{k-1} \min\{\alpha, b_i\} \geq \Omega \ . \tag{4.7}$$

The left-hand side of (4.7), as a function of α, is a piecewise-linear function of α

$$C(\alpha) = \begin{cases} k\alpha, \alpha \in [0,b_0] \\ b_0 + (k-1)\alpha, \alpha \in (b_0,b_1] \\ \vdots \\ b_0 + b_1 + \ldots + b_{k-2} + \alpha, \alpha \in (b_{k-2},b_{k-1}] \\ b_0 + b_1 + \ldots + b_{k-1}, \alpha \in (b_{k-1},\infty) \end{cases} . \tag{4.8}$$

$C(\alpha)$ is strictly increasing from 0 to its maximum $b_0 + b_1 + \ldots + b_{k-1}$ value as α increases from 0 to b_{k-1} . To find the minimum α such that $C(\alpha) \geq \Omega$, we let $\alpha^* = C^{-1}(\Omega)$ if $\Omega \leq b_0 + b_1 + \ldots + b_{k-1}$

$$\Rightarrow \alpha^* = \begin{cases} \frac{\Omega}{k}, \Omega \in [0, kb_0] \\ \frac{\Omega - b_0}{k-1}, \Omega \in (kb_0, b_0 + (k-1)b_1] \\ \vdots \\ \Omega - \sum_{j=0}^{k-2} b_j, \Omega \in (\sum_{j=0}^{k-2} b_j + b_{k-2}, \sum_{j=0}^{k-1} b_j] \end{cases} . \tag{4.9}$$

For $i = 1, \ldots .k-1$, the i-th condition in the above expression is

$$\alpha^* = \frac{\Omega - \sum_{j=0}^{i-1} b_j}{k-i}, \text{ for } \Omega \in (\sum_{j=0}^{i-1} b_j + (k-i)b_{i-1}, \sum_{j=0}^{i} b_j + (k-i-1)b_i) \ . \tag{4.10}$$

Note from definition of $\{b_i\}$ (4.6) that

$$\begin{aligned}
&\sum_{j=0}^{i-1} b_j \\
&= \sum_{j=0}^{i-1} \left(\tfrac{md-mk+mj+m}{M_R} - (m-1) \right) \gamma_R \\
&= i\gamma_R(\tfrac{md-mk+mj+m}{M_R} - (m-1)) + (\tfrac{mi(i-1)}{2M_R})\gamma_R \\
&= i\gamma_R \left(\tfrac{2md-2mk-2(m-1)M_R+mi+m}{2M_R} \right) \\
&= \gamma_R g(i)
\end{aligned} \tag{4.11}$$

and

$$\begin{aligned}
&\sum_{j=0}^{i-1} b_j + (k-i-1)b_i \\
&= \gamma_R(i+1)(\tfrac{2md-2mk-2(m-1)M_R+m(i+1)+m}{2M_R}) \\
&+(k-i-1)\gamma_R(\tfrac{2md-2mk+2mi+2m-2(m-1)M_R}{2M_R}) \\
&= \gamma_R \tfrac{i(2mk-mi-m)+k(2md-2mk+2m-2M_R(m-1))}{2M_R} \\
&= \gamma_R \tfrac{\Omega}{f(i)} \quad .
\end{aligned} \tag{4.12}$$

Then we have expression of $f(i)$ and $g(i)$ as follows :

$$f(i) = \frac{2\Omega M_R}{i(2mk-mi-m)+k(2md-2mk+2m-2M_R(m-1))} \ , \tag{4.13}$$

and

$$g(i) = i\frac{2md-2mk-2(m-1)M_R+mi+m}{2M_R} \ . \tag{4.14}$$

We use (4.11) and (4.12) to substitute into (4.10). Hence, we have another expression of (4.10) that is easier to write :

$$\alpha^* = \frac{\Omega - g(i)\gamma_R}{k-i} \ , \text{ for } \Omega \in (\frac{\Omega\gamma_R}{f(i-1)}, \frac{\Omega\gamma_R}{f(i)}) \ . \tag{4.15}$$

And we can get the relation between storage per node and remote repair bandwidth, which we will introduce in the next subsection.

4.2.3 Tradeoff Between Storage per Node and Remote Repair Bandwidth

In our optimization problem, we fix d, m, M_R, and γ_R and minimize the storage per node α. Note that parameters n, k, d, M_R are integers, and Ω, γ_R, m are real numbers.

$$\begin{aligned} &\alpha^* \triangleq \min \alpha, \\ &subject\ to: \sum_{i=0}^{k-1} \min\left\{\alpha, (\tfrac{md-mi}{M_R} - (m-1))\gamma_R\right\} \geq \Omega \end{aligned} \quad . \tag{4.16}$$

Our objective function is

$$\min \alpha \ , \tag{4.17}$$

and the constraint is

$$\sum_{i=0}^{k-1} \min\left\{\alpha, (\frac{md-mi}{M_R} - (m-1))\gamma_R\right\} \geq \Omega \ . \tag{4.18}$$

After solving this mixed integer programming problem (4.16) using result (4.15) and changing the interval according to remote repair bandwidth, we get minimum storage α^* related to remote repair bandwidth :

$$\alpha^* = \begin{cases} \frac{\Omega}{k}, \gamma_R \in [f(0), \infty) \\ \\ \frac{\Omega - g(i)\gamma_R}{k-i}, \gamma_R \in [f(i), f(i-1)) \end{cases} , \tag{4.19}$$

where

$$f(i) = \frac{2\Omega M_R}{i(2mk - mi - m) + k(2md - 2mk + 2m - 2M_R(m-1))} \tag{4.20}$$

$$g(i) = i\frac{2md - 2mk - 2(m-1)M_R + mi + m}{2M_R} \ . \tag{4.21}$$

Because the function $f(i)$ decreases while i increases, the minimum remote repair bandwidth can be obtained while $i = k - 1$. Therefore the minimum remote repair bandwidth is expressed as:

$$\gamma_{R,\min} = f(k-1) = \frac{2\Omega M_R}{k(2md - mk + m - 2mM_R + 2M_R)} \ . \tag{4.22}$$

We find two special points that represent the minimum storage and minimum remote repair bandwidth respectively. They are on the two ends of the optimal tradeoff curve (see Fig. 5.1). It can be verified that the minimum storage point is achieved by the pair

$$(\alpha_{MSR}, \gamma_{R,MSR}) = \left(\frac{\Omega}{k}, \frac{\Omega M_R}{k(md - mk + m - (m-1)M_R)}\right) \ , \tag{4.23}$$

and it also can be verified that the minimum remote repair bandwidth point is achieved by

$$(\alpha_{MBR}, \gamma_{R,MBR}) = \left(\frac{\Omega(mk + 2mM_R - 2md - 2M_R)}{k(2md - mk + m + 2M_R - 2mM_R)}, \frac{2\Omega M_R}{k(2md - mk + m - 2mM_R + 2M_R)}\right) \ . \tag{4.24}$$

Finally, based on the solution, we can find the optimal tradeoff curve of storage per node and remote repair bandwidth, and we will show the result (see Fig. 5.1).

4.3 Minumum Storage per Node Under Security Level Requirement

4.3.1 Security Parameter

In this thesis, we define λ as *security parameter*, which is the probability that user's data can be reconstructed by an eavesdropper during the repair process. Its value is between 0 to 1.

For example, we consider a bit sequence “10110100“ transmitted in the network. An eavesdropper has the probability $^1/_{2^8}$ to correctly guess the whole bit sequence. If the four left-hand side bits “1011“ are eavesdropped by the eavesdropper. The eavesdropper has the probability $^1/_{2^4}$ to correctly guess the remaining bits in the bit sequence, which is much higher than the none eavesdropping one.

We know that the remote repair bandwidth $M_R(i)\beta_R$ can be eavesdropped, so the eavesdropper can obtain $M_R(i)\beta_R$ amount of information. And the information that the eavesdropper does not know is $\Omega - M_R(i)\beta_R$. Therefore, the probability for a eavesdropper to correctly guess the remaining bits in the bit sequence is

$$\lambda = \frac{1}{2^{\Omega - M_R(i)\beta_R}} .$$

It is also the probability that he/she can know the whole bit sequence.

4.3.2 Security Level Requirement

Next, in a user’s points of view, he/she will specify a security level requirement. In this thesis, the notation is σ. It is the security rate that storage system can prevent an eavesdropper from reconstructing original data when he/she can observe remote repair bandwidth, and its value is between 0 to 1 .

The value 0 represents an eavesdropper can gather whole data object from remote repair bandwidth. On the other hand, the value 1 represents the system is perfect secrecy, which means the probability to for an eavesdropper to guess the entire original data object from remote repair bandwidth is the same as random guess. The higher value of σ means the higher security level requirement, in other words, the user asks for more secure storage service. The security level requirement must be satisfied and hence the security level provided by the storage system must be higher than or equal to σ.

To normalize the value of σ, it will be divided by $1 - 2^{-\Omega}$. So we define σ :

$$\sigma = \frac{1-\lambda}{1-2^{-\Omega}} \ . \tag{4.25}$$

In order to achieve the security level requirement, the probability for an eavesdropper not to correctly guess original data must be larger than or equal to user-specified security level requirement after normalization, i.e.,

$$\sigma \leq (1 - \frac{1}{2^{\Omega - M_R(i)\beta_R}}) \Big/ (1 - 2^{-\Omega}) \ . \tag{4.26}$$

Then we substitute σ using definition (4.25) into (4.26) and get the remote repair bandwidth upper-bound under security level requirement :

$$M_R(i)\beta_R = \gamma_R(i) \leq \Omega + \log_2\lambda \ \text{, for every } i \ . \tag{4.27}$$

Remind that $M_R(i)\beta_R$ is remote repair bandwidth for the i-th new-comer, we obtain the relation between remote repair bandwidth and security level requirement. So given Ω and σ, we can get remote repair bandwidth upper-bound. Furthermore, we imply relation between storage per node and security level requirement in the next subsection.

4.3.3 Relation Between Storage per Node and Security Level Requirement

Based on the definitions in the previous subsections, here we imply relation between storage per node and security level requirement. It is derived as bellow. And further, we can find the minimum storage per node that satisfies user's security level requirement.

Given data object size Ω and user-specified security level requirement σ. We can calculate λ from (4.25). Next, from (4.27), an eavesdropper can know $\Omega + \log_2\lambda$ amount of information at most by observing remote repair bandwidth. We define $\widetilde{\gamma_R}$ as maximum remote repair bandwidth and $\widetilde{\alpha^*}$ as minimum storage per node under the security level requirement. Thus we have $\widetilde{\gamma_R} = \Omega + \log_2\lambda$ and can find the point $(\widetilde{\gamma_R}, \widetilde{\alpha^*})$ on the tradeoff curve, and finally we find $\widetilde{\alpha^*}$ (minimum storage under security level requirement). This is our main result.

Based on (4.19), we have relation between storage per node and security level requirement.

$$\widetilde{\alpha^*} = \begin{cases} \frac{\Omega}{k}, & (\Omega + \log_2\lambda) \in [f(0), \infty) \\ \\ \frac{\Omega - g(i)(\Omega+\log_2\lambda)}{k-i}, & (\Omega + \log_2\lambda) \in [f(i), f(i-1)) \end{cases} \tag{4.28}$$

where

$$f(i) = \frac{2\Omega M_R}{i(2mk - mi - m) + k(2md - 2mk + 2m - 2M_R(m-1))} \tag{4.29}$$

$$g(i) = i\frac{2md - 2mk - 2(m-1)M_R + mi + m}{2M_R}, \tag{4.30}$$

given data object size Ω and security level requirement σ.

4.4 Upper-bound of Amount of Stored Data Under Perfect Secrecy

A user may want to store data under perfect secrecy. We further analyze upper-bound of amount of stored data under perfect secrecy. That is, the maximum amount of data that can be securely stored in the storage system without leaking any information to eavesdroppers. We want to know that whether it is possible not to leak any information to eavesdroppers. We use the concept of information theory to analyze the upper-bound as below :

Consider a distributed storage system using (n, k) code with $N_L \geq k$. Let S be a random vector uniformly distributed over ${F_q}^R$, representing the data object at the source node with $H(S) = R$. For a new-comer node x^i, let D_i and C_i be the random variables representing its downloaded data and stored content respectively. Assume that the storage nodes $x^1, x^2, ..., x^k$ have failed consecutively, and were replaced during the repair process by the nodes $x^{n+1}, x^{n+2}, ..., x^{n+k}$ respectively. Now suppose that a eavesdropper accesses nodes in $R = \left\{x^{n+1}, x^{n+2}, ..., x^{n+N_R}\right\}$ while they were being repaired, and consider a data collector connected to the nodes in $B = \left\{x^{n+1}, x^{n+2}, ..., x^{n+k}\right\}$. The reconstruction property implies $H(S|C_B) = 0$, and the perfect secrecy condition implies $H(S|D_R) = H(S)$. We can therefore write

$$\begin{aligned}
&H(S) = H(S|D_R) - H(S|C_B) \\
&\leq H(S|C_R) - H(S|C_B) \\
&= H(S|C_R) - H(S|C_{B\setminus R}, C_R) \\
&= I(S; C_{B\setminus R}|C_R) \\
&\leq H(C_{B\setminus R}|C_R) \\
&= H(C_{n+N_R+1}, C_{n+N_R+2}, ..., C_{n+k}|C_{n+1}, ..., C_{n+N_R}) \\
&= \sum_{i=N_R+1}^{k} H(C_{n+i}|C_{n+1}, C_{n+2}, ..., C_{n+i-1}) \\
&\leq \sum_{i=N_R+1}^{k} \min\{(M_L(i) - i)\beta_L + M_R(i)\beta_R, \alpha\} .
\end{aligned} \tag{4.31}$$

Therefore,

$$\sum_{i=N_R+1}^{k} \min\{(M_L(i) - i)\beta_L + M_R(i)\beta_R, \alpha\} \tag{4.32}$$

is our upper-bound of amount of stored data under perfect secrecy.

When $\sigma = 1$, then $\lambda = 2^{-\Omega}$. It means that there is no remote repair bandwidth observed by eavesdroppers. However, there always exists remote repair bandwidth in our scenario since we consider the worst case that the new-comer downloaded data from all the storage node in the remote datacenter. Therefore, the storage system can not achieve perfect secrecy (see the gap (red double arrow) in Fig. 4.3).

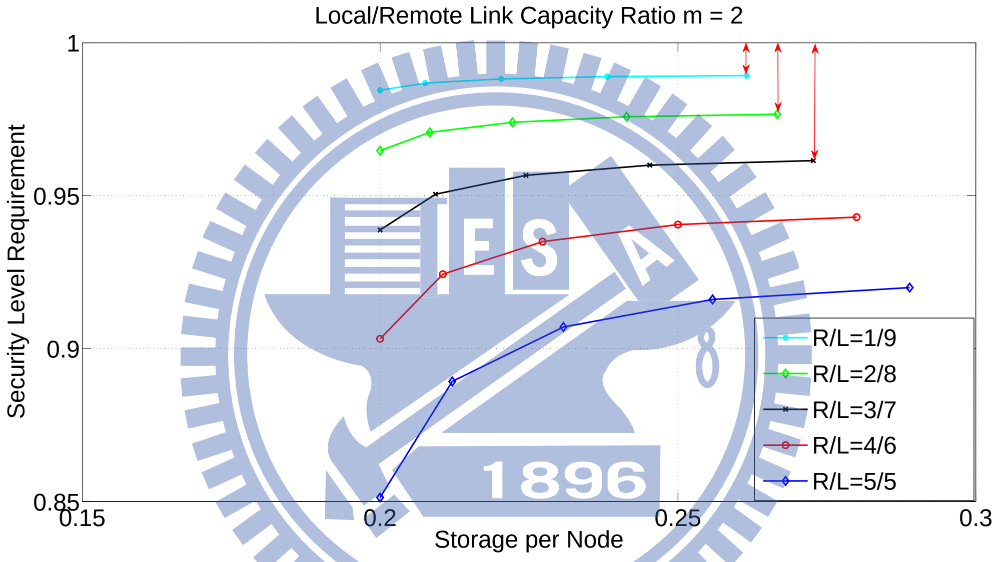

Figure 4.3: Gaps between security level of different curves and perfect secrecy condition.

CHAPTER 5

Numerical Results and Discussions

In this chapter, we show the optimal tradeoff curve between storage per node and three system parameters (remote repair bandwidth, security level requirement, data reliability). In addition, we make some discussions about the numerical results. Our common numerical result scenario parameters are defined. We analysis the optimal tradeoff curve in inter-datacenter scenario based on different number of storage nodes in the local and remote datacenter. We use $(10, 5)$ code and assume there are total ten storage nodes in the storage system, i.e. $n = 10$, $k = 5$. And the data object size Ω in our scenario is 1 . To discuss the tradeoff curves, we use five different pairs of number of local and remote storage nodes. The five pairs (N_L, N_R) are $(5, 5), (6, 4), (7, 3), (8, 2)$, and $(9, 1)$. We denote R/L as the number of storage nodes in the remote/local datacenter to make it easier to observe.

5.1 Tradeoff Curve Between Storage per Node and Remote Repair Bandwidth

In this case, we discuss the tradeoff curve between storage per node and remote repair bandwidth. We give the initial parameters of the storage system in Table 5.1. The capacity of local link is two times larger than remote link, i.e., $\beta_L = 2 * \beta_R$. The local/remote link capacity ratio m is 2.

The tradeoff curve is shown in Fig. 5.1. The different pairs of number of local and remote storage nodes are corresponding to different mark styles and colors. We have the following discussions:

- Most of all, the storage per node and remote repair bandwidth are in a tradeoff relation, that is, storage per node deceases while remote repair bandwidth increases. It is a strictly decreasing curve. The tradeoff curve changes in different number of remote and local datacenter storage nodes scenarios. The more remote storage nodes, because it causes higher remote repair bandwidth, its corresponding curve is located in the higher remote repair bandwidth value interval.

- The two special points are shown in the curve. All the curves have these two special points. They are also on the two ends of the curve. The minimum storage points in all the curves are located at value 0.2 where these points bring the maximum remote repair bandwidth.

- If we do not differentiate local and remote datacenters as the scenario in [11], the storage per node value is a constant in our result. That is, it does not change with remote repair bandwidth.

- We compare with different curves, under the same remote repair bandwidth. If data is stored in more storage nodes in remote datacenter, the storage per node is larger.

Table 5.1: Simulation Parameters for Tradeoff Curve Between Storage per Node and Remote Repair Bandwidth With Local/Remote Link Capacity Ratio $m = 2$.

Parameter	**Value**
Number of total nodes (n)	**10**
Coding parameter (k)	**5**
Original data size (Ω)	**1**
Number of the surviving nodes (d)	**9**
Amount of downloaded bits from local datacenters (β_L)	$\beta_L = m * \beta_R$
Amount of downloaded bits from remote datacenters (β_R)	**1**
Local/remote link capacity ratio (m)	**2**

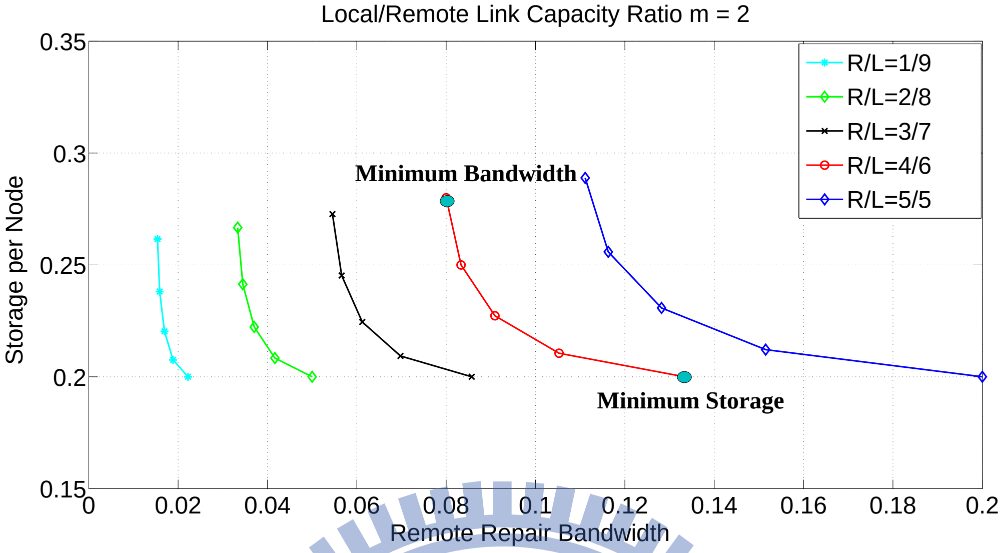

Figure 5.1: Optimal tradeoff curve between storage per node and remote repair bandwidth.

On the other hand, under the same storage per node, if data is stored in more storage nodes in remote datacenter, it can cause more remote repair bandwidth. It has higher risk to leak information.

5.2 Tradeoff Curve Between Storage per Node and Security Level Requirement

In this case, we discuss the tradeoff curve between storage per node and security level requirement. We give the initial parameters of the storage system in Table 5.2 and Table 5.3. In addition, we compare two different capacity ratios of local link and remote link. The values are 2 and 2.5 . So local/remote link capacity ratio m are 2 and 2.5 corresponding to different curves respectively.

Table 5.2: Simulation Parameters for Tradeoff Curve Between Storage per Node and Security Level Requirement With Local/Remote Link Capacity Ratios $m = 2$.

Parameter	**Value**
Number of total nodes (n)	**10**
Coding parameter (k)	**5**
Original data size (Ω)	**1**
Number of the surviving nodes (d)	**9**
Amount of downloaded bits from local datacenters (β_L)	$\beta_L = m * \beta_R$
Amount of downloaded bits from remote datacenters (β_R)	**1**
Local/remote link capacity ratio -case I (m)	**2**

Table 5.3: Simulation Parameters for Tradeoff Curve Between Storage per Node and Security Level Requirement With Local/Remote Link Capacity Ratios $m = 2.5$.

Parameter	**Value**
Number of total nodes (n)	**10**
Coding parameter (k)	**5**
Original data size (Ω)	**1**
Number of the surviving nodes (d)	**9**
Amount of downloaded bits from local datacenters (β_L)	$\beta_L = m * \beta_R$
Amount of downloaded bits from remote datacenters (β_R)	**1**
Local/remote link capacity ratio -case II (m)	**2.5**

The tradeoff curves are shown in Fig. 5.2 and Fig. 5.3, where the values of m are 2 and 2.5 respectively. We have the following discussions:

- Most of all, the storage per node and security level requirement are in a tradeoff relation, that is, storage per node increases while security level requirement increases. It is a strictly increasing curve. We can imply that increasing storage space cost can improve the security level requirement. Under the same link capacity ratio, the tradeoff curve changes in different number of remote and local datacenter storage nodes scenarios. The less remote storage nodes, its corresponding curve is located in the higher security level requirement value interval. Because it causes lower remote repair bandwidth in the case with less remote storage nodes, it can achieve higher security level requirement.

- In Figure 5.2, we can see all curves have vertical asymptotic lines. It has the same phenomenon in Figure 5.3. The vertical asymptotic lines represent the maximum security level that the storage system can achieve. All curves cannot exceed the vertical asymptotic lines. Because there are always information transmitted via remote repair bandwidth which can be observed by eavesdroppers, the security level cannot achieve 1 .

- Compared with different curves in same link capacity ratio, (e.g. Fig. 5.2), to achieve the same security level requirement, it cost more storage space in the case with more remote storage nodes. Because it causes more remote repair bandwidth with more remote storage nodes, it has to cost more storage space to achieve the same security level requirement compared with the less remote storage nodes one.

- We compare the curves with different link capacity ratios (see Fig. 5.4). Under the same security level requirement and the same pair of number of storage nodes in local/remote datacenter. If the link capacity ratio is larger, which means local link

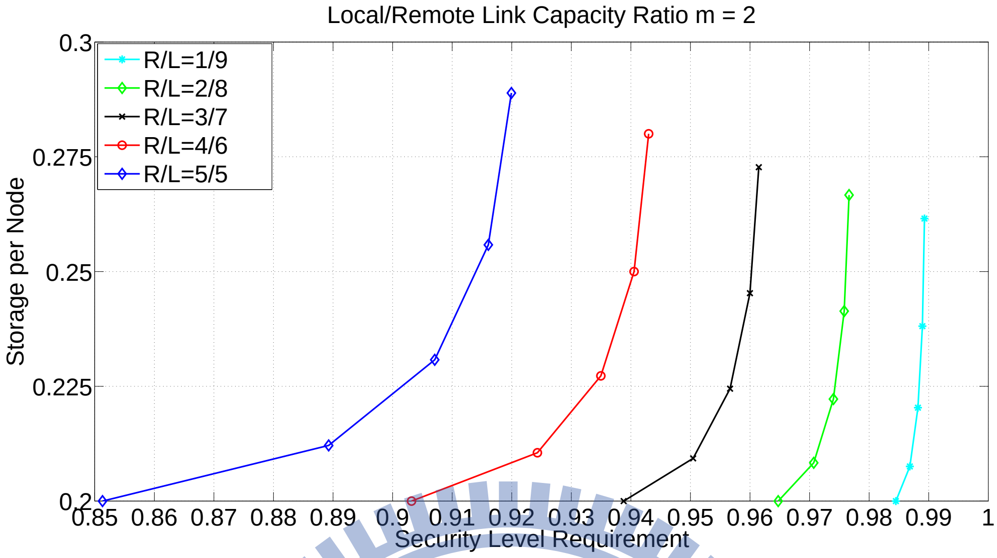

Figure 5.2: Tradeoff between storage per node and security level requirement with local/remote link capacity ratios $m = 2$.

capacity is much larger than remote link capacity, the storage cost is less. Because larger capacity ratio causes lower remote repair bandwidth in the case with larger link capacity ratio, it can achieve the same security level with less storage space. On the other hand, under the same storage cost and the same pair of number of storage nodes in local/remote datacenter. If the link capacity ratio is larger, it can achieve higher security level, and hence have lower risk to leak information.

5.3 Tradeoff Between Storage per Node and Data Reliability

In this case, we find another tradeoff between storage per node and data reliability in addition, and discuss this tradeoff. We give the initial parameters of the storage system

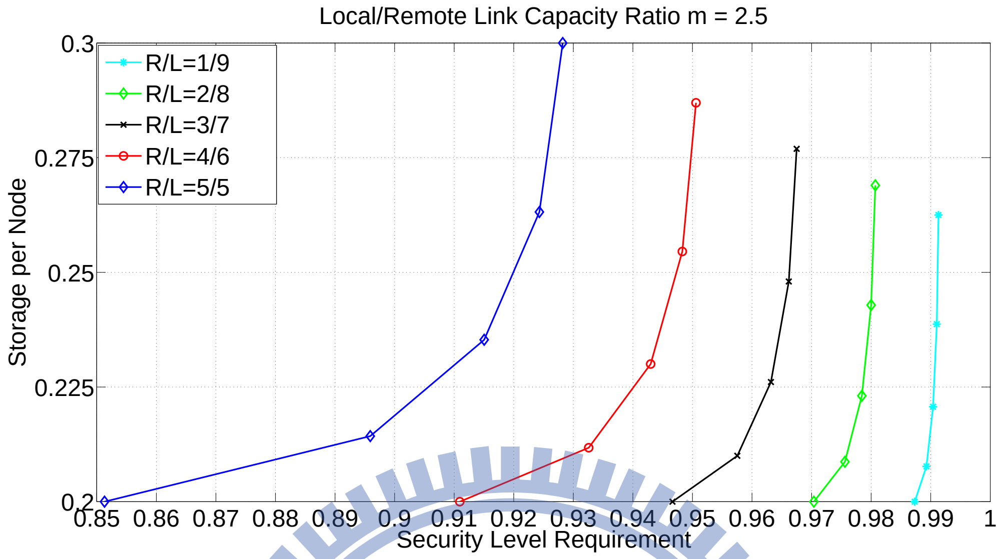

Figure 5.3: Tradeoff between storage per node and security level requirement with local/remote link capacity ratios $m = 2.5$.

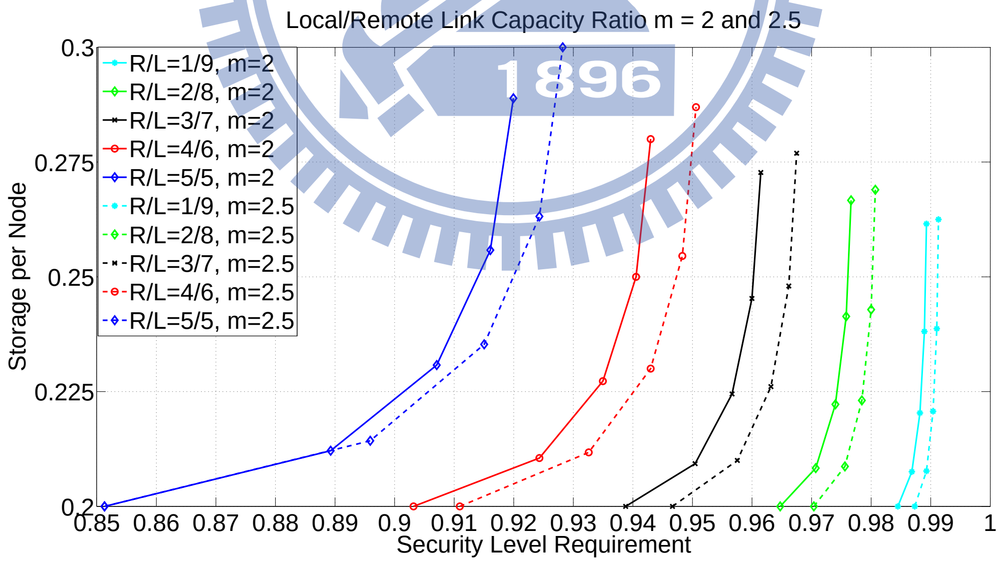

Figure 5.4: Comparison between different capacity ratios of local link and remote link .

in Table 5.4. We find this tradeoff relation in the teadeoff curve of storage per node and security level requirement. The local/remote link capacity ratio m is 2 .

Figure 5.5 shows the tradeoff relation for $m = 2$ and $\Omega = 1$. We also have the following discussions:

- We know that if data are stored in less centralized way (e.g. $(R/L) = (5, 5)$), it needs larger storage per node to satisfy the security requirement. We use an example (see Fig. 5.5) to illustrate that there is a tradeoff between storage cost and reliability for different number of remote and local storage nodes. Given a security level requirement, we can find two different storage schemes that satisfy the security level requirement (0.91 in the example) easily. We have $R/L = 4/6$ and $R/L = 5/5$ for the minimum storage and the maximum reliability, respectively, since the data stored using the latter storage scheme can be recovered if the entire nodes in the local datacenter are failed (such as a fire disaster) whereas the former cannot.

- It is also an interesting issue about the tradeoff between storage cost and data reliability under same security level requirement in network coding based distributed storage systems in inter-datacenter scenario. How to analyze the data reliability is full of different points of view, we will leave it as future discussion issue.

Table 5.4: Simulation Parameters for Tradeoff Curve Between Storage per Node and Data Reliability With Local/Remote Link Capacity Ratio $m = 2$.

Parameter	Value
Number of total nodes (n)	10
Coding parameter (k)	5
Original data size (Ω)	1
Number of the surviving nodes (d)	9
Amount of downloaded bits from local datacenters (β_L)	$\beta_L = m * \beta_R$
Amount of downloaded bits from remote datacenters (β_R)	1
Local/remote link capacity ratio -case I (m)	2

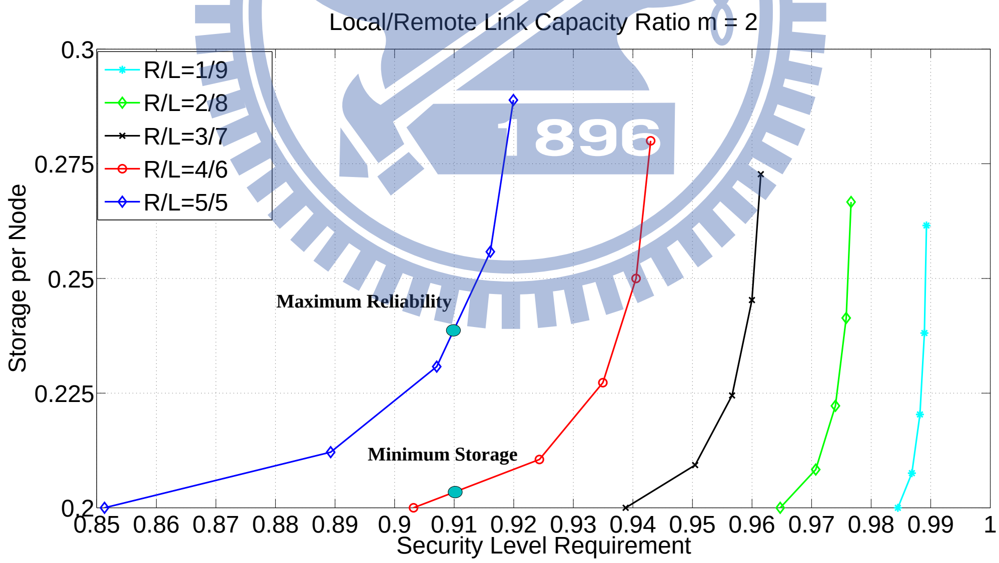

Figure 5.5: Tradeoff between storage per node and data reliability with local/remote link capacity ratio $m = 2$.

CHAPTER 6

Conclusions

6.1 Tradeoff Curve Between Storage per Node and Remote Repair Bandwidth

We have investigated the link eavesdropping problem for repairing network coded data from remote distributed storage. We first presented the information flow graph analysis and showed the fundamental tradeoff curve of remote repair bandwidth and storage per node. Then we derived the minimum storage per node for achieving required security level. Finally, we found that there exist another tradeoff for storage cost and reliability for different number of remote and local storage nodes in the considered scenario. This work is a first step towards understanding the security of distributed storage with inter-datacenter communication. In the future, we will focus on the optimal allocation problem for such system with the consideration of storage cost, security and reliability.

6.2 Future Research

For the future research of the thesis, we provide the following suggestions to extend our work in distributed storage with inter-datacenter communication :

The consideration of storage cost, security and reliability in cloud datacenter is important. How to find the relation between them, make a best allocation choice for users, and optimal revenue for storage system providers will be an important issue in the future.

Bibliography

[1] Q. He, Z. Li, and X. Zhang, "Study on cloud storage system based on distributed storage systems," *IEEE International Conference on Computational and Information Sciences*, pp. 1332–1335, Dec. 2010.

[2] S. Rhea, C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz, "Maintenance-free global data storage," *IEEE Internet Computing*, vol. 5, no. 5, pp. 40–49, Sep. 2001.

[3] R. Bhagwan, K. Tati, Y.-C. Cheng, S. Savage, and G. M. Voelker, "Total recall: System support for automated availability management," *Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation (NSDI)*, vol. 1, pp. 25–25, 2004.

[4] F. Dabek, J. Li, E. Sit, J. Robertson, M. F. Kaashoek, and R. Morris, "Designing a dht for low latency and high throughput," *Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation (NSDI)*, pp. 85–98, 2004.

[5] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decentralized erasure codes for distributed networked storage," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2809–2816, 2006.

[6] Y. Singh, F. Kandah, and W. Zhang, "A secured cost-effective multi-cloud storage in cloud computing," *IEEE Conference on Computer Communications Workshops*, pp. 619–624, 2011.

[7] R. Bhagwan, D. Moore, S. Savage, and G. M. Voelker, "Replication strategies for highly available peer-to-peer storage," *Future directions in distributed computing*, pp. 153–158, 2003.

[8] R. Rodrigues and B. Liskov, "High availability in dhts: Erasure coding vs. replication," *Peer-to-Peer Systems IV 4th International Workshop IPTPS*, pp. 226–239, 2005.

[9] H. Weatherspoon and J. Kubiatowicz, "Erasure coding vs. replication: A quantitative comparison," *Peer-to-Peer Systems 1st International Workshop (IPTPS)*, pp. 328–337, 2002.

[10] Y. Hu, C.-M. Yu, Y. K. Li, P. P. Lee, and J. C. Lui, "On the practicality and extensibility of a network-coding-based distributed file system," *IEEE International Symposium on Network Coding (NetCod)*, pp. 1–6, 2011.

[11] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[12] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 476–489, 2011.

[13] Q. Yu, K. W. Shum, and C. W. Sung, "Minimization of storage cost in distributed storage systems with repair consideration," *Global Telecommunications Conference (GLOBECOM)*, pp. 1–5, 2011.

[14] S. Pawar, S. El Rouayheb, and K. Ramchandran, "On secure distributed data storage under repair dynamics," *IEEE International Symposium on Information Theory Proceedings (ISIT)*, pp. 2543–2547, 2010.

[15] N. B. Shah, K. Rashmi, and P. V. Kumar, "Information-theoretically secure regenerating codes for distributed storage," *IEEE Global Telecommunications Conference*, pp. 1–5, 2011.

[16] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems from malicious nodes," *IEEE International Symposium on Information Theory Proceedings*, pp. 1452–1456, 2011.

[17] P. F. Oliveira, L. Lima, T. T. Vinhoza, J. Barros, and M. Médard, "Trusted storage over untrusted networks," *IEEE Global Telecommunications Conference*, pp. 1–5, 2010.

[18] N. Shah, K. Rashmi, P. Kumar, and K. Ramchandran, "Regenerating codes for distributed storage networks," *Arithmetic of Finite Fields 3rd International Workshop (WAIFI)*, pp. 215–223, 2010.

[19] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[20] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[21] W. Qiao, J. Li, and J. Ren, "An efficient error-detection and error-correction (edec) scheme for network coding," *IEEE Global Telecommunications Conference*, pp. 1–5, 2011.

[22] N. Cai and R. W. Yeung, "Network coding and error correction," *Proceedings of the IEEE Information Theory Workshop*, pp. 119–122, 2002.

[23] J. P. Vilela, L. Lima, and J. Barros, "Lightweight security for network coding," *IEEE International Conference on Communications*, pp. 1750–1754, 2008.

[24] T. K. Dikaliotis, A. G. Dimakis, and T. Ho, "Security in distributed storage systems by communicating a logarithmic number of bits," *IEEE International Symposium on Information Theory Proceedings*, pp. 1948–1952, 2010.

[25] K. Rashmi, N. B. Shah, K. Ramchandran, and P. V. Kumar, "Regenerating codes for errors and erasures in distributed storage," *IEEE International Symposium on Information Theory Proceedings*, pp. 1202–1206, 2012.

[26] Y. Wu, A. Dimakis, and K. Ramchandran, "Deterministic regenerating codes for distributed storage," in *Allerton Conference on Control, Computing, and Communication*. Citeseer, 2007.

[27] S. Jaggi and M. Langberg, "Resilient network codes in the presence of eavesdropping byzantine adversaries," *IEEE International Symposium on Information Theory*, pp. 541–545, 2007.

[28] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, "Resilient network coding in the presence of byzantine adversaries," *IEEE International Conference on Computer Communications*, pp. 616–624, 2007.

[29] T. Ernvall, S. E. Rouayheb, C. Hollanti, and H. V. Poor, "Capacity and security of heterogeneous distributed storage systems," *arXiv preprint arXiv:1211.0415*, 2012.

[30] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems against eavesdropping and adversarial attacks," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6734–6753, 2011.

Vita

Chen-Hung Liao was born in Taiwan, R. O. C. in 1988. He received a B.S. in Computer Science from National Chiao-Tung University in 2011. From July 2011 to September 2013, he worked his Master degree in the Mobile Communications and Cloud Computing Lab in the Institute of Computer Science and Engineering at National Chiao-Tung University. His research interests are in the field of Cloud Computing.