

行政院國家科學委員會補助專題研究計畫 成果報告
 期中進度報告

分散式網路儲存系統安全傳輸問題的研究

Security issues of distributed networked storage systems

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC 98-2221-E-009-068-MY3

執行期間：98年8月1日至101年7月31日

第三年度：100年8月1日至101年7月31日

計畫主持人：曾文貴 教授

計畫參與人員：林孝盈、官正傑、林輝讓、劉正偉、劉麗君

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、
列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

執行單位：國立交通大學 資訊工程系

中華民國 101 年 10 月 28 日

中文摘要

本研究計畫將研究分散式網路儲存系統的安全儲存機制。網路儲存系統提供使用者儲存資料在網路上的儲存系統中，再透過網路進行資料存取。目前的分散式網路儲存系統首要注重的是效率，其次才是安全性，我們認為在資料隱私性上還有許多改善的空間。

第一年度（98-99）我們發展了一個以隨機線性編碼基礎的安全分散式網路儲存系統。在我們的系統中，資料透過公開金鑰系統加密來達到高度資料隱私性，隨機線性編碼方法則是提供了儲存系統的容錯能力。整體系統的運作符合分散式系統的環境特質，論文已在 IEEE TPDS（2010）期刊發表。

第二年度（99-100）我們基於去年發展的安全儲存系統，繼續提供多樣性的功能，例如，如何將安全儲存的資料送給第三者（forwarding），資料擁有者不需將儲存的資料取回解密後再上傳，這樣可以減少大量的頻寬使用。論文成果已經在 IEEE TPDS（2012）上發表。

第三年度（100-101），我們基於先前兩年的成果，有一個安全強固且具有資料傳送給第三者的分散式雲端儲存系統上，提出當系統的一些伺服器出現錯誤時可以修復的機制，成果發表在 IEEE TrustCom-2011 會議上，完整論文也已投稿到知名期刊。

關鍵詞：分散式網路儲存系統，公開金鑰加密，隨機容錯編碼，資料安全傳送，資料傳送。

英文摘要

In this project, we study security issues of distributed networked storage systems. A networked storage system enables users to store data and to access data via Internet access.

Currently, distributed networked storage systems are designed for efficiency and security is a second issue. One of goals of this research is to improve the data confidentiality in distributed networked storage systems.

In the first year (2009-2010), we developed a random linear code-based secure distributed networked storage system. The system uses a public key encryption scheme to provide high data confidentiality and uses a random linear code to achieve the data robustness. The data storing and retrieval processes are fully distributed. The paper has been published in IEEE TPDE.

In second year (2010-2011), we develop the system such that it can support the functionality of data forward. In this system, the data owner can securely forward the stored data in the distributed storage system to another user. The owner does not need to retrieve the data back to process it for forwarding to another user. The owner simply sends a proxy re-encryption key to the storage servers and the servers re-encrypt the data into a ciphertext that can be decrypted by the target user. This method reduces the bandwidth requirement dramatically. We have finished a manuscript and submitted it to an international journal.

In the third year (2011-2012), we continue to research on distributed storage systems. Based on the results of the previous two years, we consider repair mechanisms for our robust and secure storage system. We propose cooperative and non-cooperative repair mechanisms. The results have been published in the conference IEEE TrustCom-2011. The complete paper is submitted to a prestigious journal.

Keywords: Distributed networked storage system, public key cryptosystem, random erasure code, data forwarding, repair mechanism.

一. 計畫緣起及目的

由於高速網路與多樣隨身上網裝置的普及化，許多服務都透過網路來傳遞。較為常見的有網路信箱，搜尋引擎，網路聊天室，網路文件編輯器等。這些透過網路提供的服務系統底層都使用到了網路儲存系統的建構。我們考慮分散式網路儲存系統這個基礎服務。

一個分散式網路儲存系統包含許多儲存伺服器，彼此透過網路進行連結，其中並沒有一個長駐的中央控制管理單位，這使得整體系統較為彈性且不會在中央控制管理單位造成系統效能瓶頸，但是相對地，管理效率就叫無法掌控。

將資料儲存在網路系統中首先會面臨的問題是資料是否能夠正常取回，這點主要是透過容錯機制來防止任何系統內部的意外錯誤，另外一個新興的使用者疑慮是資料隱私性的問題，資料存放在網路儲存系統之後是否會被惡意人士竊取再利用，我們需要一個能夠同時處理系統容錯與資料隱私性的雲端儲存系統。

最基本的容錯技術就是儲存副本，像是磁碟陣列 RAID-1 或早期許多分散式儲存系統，都使用副本技術。副本技術需要付出很大的儲存成本。為了解決這個問題，Erasure codes 被提出可以應用到容錯儲存上。

RAID-5 與 RAID-6 就應用了 Erasure codes 的技術，Lincoln erasure codes 是一個特殊的 erasure codes 並被應用到儲存系統中以提供容錯能力。其他種類的 Erasure Codes 還有很多，例如 Low density parity checking codes 或者是 Evenodd codes 與 STAR codes，也都被

應用到儲存系統中來提供容錯能力。Random liner codes 可以容忍大量的儲存毀損且儲存成本較副本技術低許多，但是需要使用較多的時間進行編碼與解碼的運算。2006 年，Dimarks 與 Prabhakaran 等學者應用 random linear code 在分散式網路儲存環境中，以獲得具有容錯能力但儲存空間成本較低的儲存系統，他們的結果亦應用到無線感測網路系統中，可知在儲存空間成本上是很有效率的。

想要保障使用者的資料不被第三者得知，除了好的雲端管理與存取控管機制外，大概能做的是把資料加密後再存入系統。加密可以由雲端儲存系統來做，1993 年 Blaze 所提出的 CFS，與其衍生的 TCFS 與 NCryptfs，較為近期的系統則有 OceanStore, Plutus, 與 Tahoe。然而我們想要探討的資料隱私性，不僅是抵擋外來的攻擊，更要預防雲端儲存中的惡意主機。對使用者來說，全面相信雲端中的所有主機是較不實際的假設，如果能夠達到在使用者不用信任這些主機的情況下，仍能保障資料隱私性，這樣的保護機制與儲存系統才能真正被使用者信任，進而使用。

我們結合了 random linear code 與公開金鑰加密系統兩大工具，設計了一個安全的分散式網路儲存系統。我們的分散式儲存系統同時具有容錯能力與高度資料隱私性，除了儲存服務之外，我們也新增了金鑰管理服務以降低使用者管理金鑰上的風險。除此之外，我們還考慮如何有效率的運用儲存的資料，雖然將資料加密儲存可以提供好的安全保護，但是也限制了它們的使用，大約是使用者將資料取回解密後處理，這樣的動作需花費大量的網路頻寬，不方便且沒有效

率。如何提供有效率使用安全儲存資料的方法是研究的重點。

當資料以分散式的方式儲存在儲存伺服器時，可能損毀或遭到破壞，當這些錯誤發生時，如何利用儲存在其他伺服器的資料將錯誤的伺服器修護或對新加入的伺服器寫入一些資料，使得整個系統還具有強固與安全的特性，是值得研究的課題。

二. 研究成果

第一年度 (2009-2010)

第一年度的研究成果為提出一個安全的分散式網路儲存系統。我們的系統有三個角色，儲存伺服器，金鑰管理伺服器與使用者。假定系統有 n 個儲存伺服器， m 個金鑰管理伺服器，使用者要儲存 k 筆資料。使用者的資料將被加密後存入系統，系統會透過分散式容錯編碼 (decentralized erasure coding) 將資料分散儲存在 v 個儲存伺服器中，當使用者要將資料取回時，系統中的金鑰管理伺服器會與 u 個儲存伺服器聯絡取得資料並協助使用者進行解密運算，使用者自己再進行解碼以拿到資料。在這期間，由於儲存伺服器與金鑰伺服器都是獨立進行編碼與協助解密的程序，所以不需要一個中央控制單位的協助。

在功能性上，我們透過錯誤更正碼儲存來因應系統中儲存伺服器可能意外地斷線或儲存設備的毀損，使得系統在發生意外狀況時仍能夠提供服務。在資料隱私性上，我們則是考慮一個高度隱私性的要求，使用者的資料不僅僅是其他系統中使用者無法接觸，負責提供服務的儲存伺服器本身亦無法得知資料的內容。

研究成果的主要貢獻，從學術理論上來看，我們提供了一個結合了容錯技術與公開金鑰加密系統的密碼學工具，這個工具能夠在一個非集中式的儲存系統環境中被使用，使得系統同時具有資料可信賴與高度隱私性並且兼顧了分散式的優點，另外針對系統中資料儲存的取回正確率上，我們亦提供了一個完整的分析方式並建議了一組通用的系統參數。

從儲存系統發展與應用上來看，我們強調了資料隱私性在雲端儲存系統上的重要性與一個強度上的分野，早期網路儲存系統的隱私性是建立在完全信任儲存伺服器的假設下，僅對登入的使用者進行身分認證，我們則是強調資料隱私性的強度應該要能夠消除對儲存伺服器的信任的假設條件。

在容錯能力上來說，我們的系統能夠容忍 $(n-k)$ 個儲存伺服器錯誤與 $(m-t)$ 個金鑰管理伺服器錯誤。只要有 k 個儲存伺服器與 t 個金鑰管理伺服器仍正常運作，則使用者可以有很高的機率將資料取回。

在資料隱私性方面，因為資料都是以加密的型態被儲存，所以即使是所有的儲存伺服器都被攻擊者控制，資料內容仍能保密。我們對於金鑰管理伺服器則有較高的信任要求，我們假設這些金鑰管理伺服器有較好的安全機制以保障使用者的各個部分解密金鑰。

第二年度 (2010-2011)

為了在分散式安全的儲存系統上達到具有 data forwarding 的能力，我們提出了新的門檻式的再加密協定 (threshold re-encryption scheme)，然後將整合到安全的儲存系統裡。結合的系統具有安全、容錯、data forwarding 的

功能，這項工作的主要困難度在於如何在加密的系統上同時做容錯計算與 data forwarding。

我們將伺服器分為儲存伺服器與金鑰伺服器，其中金鑰伺服器位於私有雲中，我們將金鑰分由金鑰伺服器持分，當使用者要取回資料時，由金鑰伺服器向儲存伺服器要求資料做部分解密，當使用者有足夠的解密資料就可以將真正的資料計算出來。我們還改進了先前對儲存伺服器數 n ，分配的訊息數 v ，文件的的分割數 k 等作了更精確的計算，得到較好的 bounds。

詳細內容請見我們所附的論文。

第三年度 (2011-2012)

我們基於先前兩年的成果，有一個安全強固且具有資料傳送給第三者的分散式雲端儲存系統上，提出當系統的一些伺服器出現錯誤時可以修復的機制。我們有兩種修復機制，第一種是新加入的儲存伺服器間不相互傳遞訊息，第二種是新加入的儲存伺服器間可以相互傳遞訊息。原先對修復機制有一個最底下界值(lower bound)，用我們的方法可以得到在平均下，可以打破此下界值，在絕大多數的情形下，加入的伺服器可以跟少於 k 個原先存在的儲存伺服器溝通交換訊息，而系統還是可以保持良好的強固性。

這部分的成果發表在 IEEE TrustCom-2011 會議上，完整論文也已投稿到知名期刊。

三. 計畫成果自評

整個三年計劃我們已經發表了以下的論文：

1. Hsiao-Ying Lin, Wen-Guey Tzeng, Shiu-an-Tzuo Shen and Bao-Shuh P. Lin. A Practical Smart Metering System Supporting Privacy Preserving Billing and Load Monitoring. In the 10th International Conference on Applied

Cryptography and Network Security (ACNS 2012), June 2012.

2. Hsiao-Ying Lin, John Kubiawicz and Wen-Guey Tzeng. A Secure Fine-Grained Access Control Mechanism for Networked Storage System. In the Sixth IEEE International Conference on Software Security and Reliability (IEEE SERE 2012), June 2012.
3. Hsiao-Ying Lin, Wen-Guey Tzeng. A Secure Erasure Code-based Cloud Storage System with Secure Data Forwarding, IEEE Transactions on Parallel and Distributed Systems 23(6), pp.995-1003, 2012.
4. Hsiao-Ying Lin, Wen-Guey Tzeng, Bao-Shuh Lin. A Decentralized Repair Mechanism for Decentralized Erasure Code based Storage Systems. In the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom-2011), Nov, 2011.
5. Hsiao-Ying Lin, Wen-Guey Tzeng. A Secure Decentralized Erasure Code for Networked Storage Systems, IEEE Transactions on Parallel and Distributed Systems, 21(11), pp.1586-1596, 2010.

其中有兩篇高水準的期刊論文，另外一篇正在投稿中，研究成果符合計劃的預期。

A Decentralized Repair Mechanism for Decentralized Erasure Code based Storage Systems

Hsiao-Ying Lin*, Wen-Guey Tzeng[†], Bao-Shuh Lin*

*Intelligent Information and Communications Research Center, [†]Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan

hsiaoying.lin@gmail.com, wgtzeng@cs.nctu.edu.tw, bplin@mail.nctu.edu.tw

Abstract—Erasure code based distributed storage systems provide data robustness by storing encoded-fragments over servers. To maintain data robustness, a repair mechanism recovers a storage system from server failures by repairing encoded-fragments. For decentralized erasure code based storage systems, we propose a decentralized repair mechanism. Our mechanism has the following features. Firstly, an encoded-fragment is replenished by a combination of a number u of encoded-fragments that are randomly chosen. Secondly, the number u depends on the number of the available encoded-fragments and is independent of the pattern of missing encoded-fragments. Thirdly, multiple encoded-fragments are simultaneously replenished in parallel. We measure the communication cost in terms of the number u of required network connections for replenishing an encoded-fragment. We then conducted a numerical analysis by using traces of real systems. We find that our requirement on u is smaller than that from existing methods. Both theoretical and numerical results show that our decentralized repair mechanism outperforms existing ones in terms of the communication cost under the same consideration of efficiency cost for storage.

Keywords-decentralized erasure codes; regenerating codes; network coding; distributed storage;

I. INTRODUCTION

Erasure code based distributed storage systems provide data robustness by storing encoded-fragments over servers. An (n, k) erasure code encodes a message of k symbols to a codeword of n symbols such that the message can be decoded from any k codeword symbols. The code tolerates $n - k$ erasure errors. To store a message in an (n, k) -erasure code based distributed storage system with n servers, the message is encoded into a codeword by the erasure code and each of its codeword symbols is stored in a different server. A server failure corresponds to an erasure error of the stored codeword symbol. As long as k servers are available, the message can be recovered. In this paper, we sometimes refer a codeword symbol as an encoded fragment and use them interchangeably.

A decentralized erasure code is an erasure code that independently computes each codeword symbol for a message. Thus, the encoding process for a message consists of n parallel tasks of generating codeword symbols. Each server executes one task to compute a codeword symbol. This kind

of systems is suitable for decentralized environments, where no centralized authority coordinates the tasks, such as peer-to-peer and ad-hoc networks. Parallel computing also speeds up the storing process.

Maintenance of robustness in an erasure code based distributed storage system requires to replenish codeword symbols when servers fail or leave the system. A straightforward solution is to compute the original message from available codeword symbols and then to regenerate missing codeword symbols from the message. This approach leads to higher communication and computation cost. Another approach is to generate codeword symbols by directly combining u available ones. When a new server joins the system, it queries u available servers to generate a codeword symbol. The generated codeword symbol can be different from the missing one. But, the property that any k codeword symbols can recover the message remains.

In previous studies, efficiency is measured by the storage cost (the number of bits a server stores) and the repair bandwidth (the number of bits a new server received for replenishing a codeword symbol). However, in considering the communication cost, the cost of establishing network connections is significant. Establishing network connections between servers involves authentication and negotiation process. The entailed communication cost is significant, especially when u is large. For example, when $u = n - 1$, a new server needs to connect all available servers in the system. Thus, we measure the communication cost by the number u of required network connections, as well as the repair bandwidth.

We study repair mechanisms for decentralized erasure code based storage systems. In a decentralized erasure code based storage system, we show that $u = k$ is a sufficient condition for a repair mechanism. Specifically, we are interested in finding out whether u can be smaller than k .

Contributions. We propose a decentralized repair mechanism for decentralized erasure code based storage systems with the following features:

- A codeword symbol is replenished by a combination of a number u of randomly chosen codeword symbols without recreating the original message.

- The number u depends on the number of available codeword symbols and is independent of the pattern of missing codeword symbols.
- Multiple codeword symbols can be independently replenished.

We theoretically study the lower bound for u . The bound depends on the number of available servers and the parameter k . With a fixed k , the larger the number of available servers is, the smaller u can be. It shows flexibility between the parameter u and the number of available servers. We then conducted a numerical analysis by using traces of real systems. Both theoretical and numerical results show that u can be smaller than k . When $u < k$, the average repair bandwidth for a server failure is less than the size of the original message. From the aspect of information theory, it gives a light data confidentiality, which is independently interesting. When a new server joins the system and tries to recover a missing codeword symbol, some codeword symbols are sent to the new server from remaining servers. An eavesdropper may eavesdrop the transmitted codeword symbols and recover the original message. If u is smaller than k , the information in the eavesdropped codeword symbols is not enough to compute the message. This confidentiality is light since increasing eavesdropped codeword symbols will eventually reveal the message. Thus, it is advised to encrypt communication channel between servers.

We compare our decentralized repair mechanism with other mechanisms in terms of communication cost and storage cost. The result shows that our decentralized repair mechanism outperforms existing ones in terms of the communication cost under the same consideration of efficiency cost for storage.

II. RELATED WORK

We briefly review repair mechanisms of erasure code based distributed storage systems.

In erasure code based distributed storage systems, repairing codeword symbols is essential to maintain robustness against server failures. Since regenerating codeword symbols after reconstructing the message is costly in terms of communication and computation cost, a hybrid approach is proposed [1]. A storage server stores the message whereas other storage servers store encoded-fragments. When some servers fail, the storage server storing the message regenerates missing encoded-fragments. The asymmetric storing structure complicates system management.

Dimakis et al. introduced regenerating codes [2]. The codes are to minimize storage cost and repair bandwidth. They showed that repair bandwidth can be decreased by letting a new server query more than k servers. However, storage cost would slightly increase. The tradeoff between storage cost and repair bandwidth is described as a curve where two extreme points are highlighted. By the points, they proposed two repair mechanisms, minimum storage

regime and minimum bandwidth regime. In the minimum storage regime, a new server queries $k + 1$ randomly chosen servers; in the minimum bandwidth regime, a new server queries $n - 1$ randomly chosen servers. By using the cut-set bound of network coding in an information flow graph, a repair mechanism corresponding to a point on the curve is proved that after a system is repaired, a user retrieves a message with probability 1. More constructions and discussions of regenerating codes can be found in [3], [4].

Rashmi et al. [5] proposed exact regenerating codes, which exactly regenerate missing codeword symbols. Shah et al. [6] took the consideration that traffic conditions vary among different links. They proposed flexible regenerating codes, which allow a new server download different amounts of data from different servers. Alternative models of repair mechanisms [7], [8], [9] are proposed for different scenarios. Nevertheless, the family of regenerating codes handles only the case of one server failure. Once a server fails or leaves the system, the repair mechanism is immediately executed. This approach increases system load.

Hu et al. [10] proposed a mutually cooperative recovery mechanism to recover distributed storage systems from multiple server failures. The mechanism has two communication phases. First, each new server queries all remaining servers. Second, each new server communicates with all other new servers. Thus, a new server totally queries $n - 1$ servers. Recently, Oggier and Datta [11] proposed self-repairing homomorphic codes for repairing multiple server failures. Each new server queries a fixed number of servers to regenerate missing codeword symbols and the number can be less than k . However, a new server has to query a specific subset of old servers to regenerate some codeword symbol. There is a mapping from a codeword symbol to specific subsets of old servers for regenerating the codeword symbol. Thus, self-repairing homomorphic codes need a central table for these mappings. The deterministic self-repairing homomorphic codes are not suitable for decentralized environments.

Dikaliotis et al. [12] studied the method of detecting faulty errors in distributed storage systems. Rashmi et al. [13] proposed a framework that integrates two erasure codes to obtain features from both codes. Pawar et al. [14] discussed data confidentiality issue when a repair mechanism is executed. Papailiopoulos and Dimakis [15] gave a reduction between the problem of maximizing data confidentiality and the problem of minimizing repair bandwidth.

III. OUR REPAIR MECHANISM

We firstly describe a decentralized erasure code based storage system as our system model and then introduce our repair mechanism. We show our bound on the parameter u for the repair mechanism.

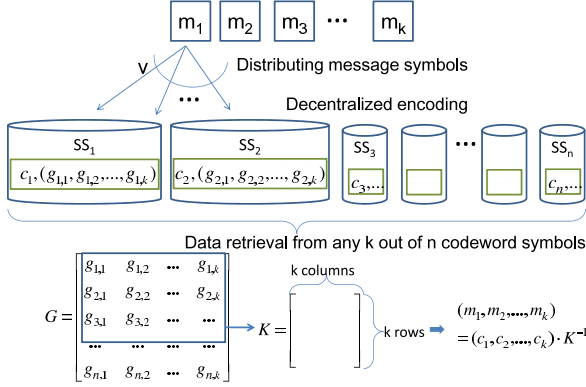


Figure 1. The model of decentralized erasure code based storage systems.

A. Decentralized Erasure Code based Storage System

Dimakis et al. [16] proposed a decentralized erasure code based storage system where the encoding process is accomplished by decentralized servers in parallel. Afterward, for strengthening data confidentiality, Lin and Tzeng [17], [18] proposed secure decentralized erasure codes where data are encoded in an encrypted form. Illustrated in Fig. 1, a decentralized erasure code based storage systems is described as follows. There are n servers, $\text{SS}_1, \text{SS}_2, \dots, \text{SS}_n$, and a message is represented as a vector of symbols m_1, m_2, \dots, m_k in some finite field. To store the message, each symbol is distributed to v randomly chosen servers. A server SS_i then picks a random coefficient $g_{i,j}$ for a received message symbol m_j and linearly combines all received message symbols as a codeword symbol c_i . If m_j is not received, $g_{i,j}$ is set to 0. Note that the combination is operated in the finite field. Globally, all chosen coefficients form a generator matrix $G = [g_{i,j}]$, $1 \leq i \leq n, 1 \leq j \leq k$, which encodes the vector of k message symbols to the vector of n codeword symbols. To retrieve the message, a user queries k randomly chosen servers to get k codeword symbols, say c_1, c_2, \dots, c_k , and the corresponding coefficients. The coefficients form a square matrix K , which is a submatrix of G . The user decodes the message by computing $(c_1, c_2, \dots, c_k) \times K^{-1}$, where K^{-1} is the inverse matrix of K . A successful data retrieval of the system is the event that K is invertible. The probability of a successful data retrieval is overwhelming when v is sufficiently large [16], [17], [18].

From the results in [16], the system parameters are suggested as follows in order to guarantee a high probability of a successful data retrieval. When $n = ak$, $v = b \ln k$, and $b > 5a$ with constants a and b , the probability of a successful data retrieval is at least $1 - k/p - o(1)$, where p is the prime order of the underlined group. Later in [18], these parameters are generalized for $n = ak^c$ and $c \geq 1$. When $n = ak^c$, $v = bk^{c-1} \ln k$, $b > 5a$, and $c \geq 1$ with constants a and b , the probability of a successful data retrieval is at

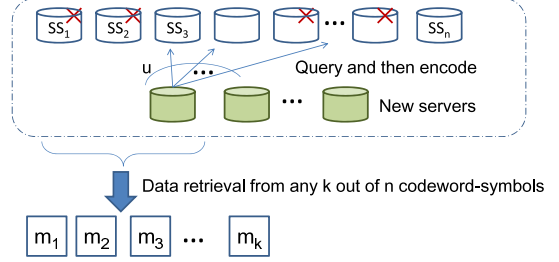


Figure 2. Our repair model for decentralized erasure code based storage systems.

least $1 - k/p - o(1)$.

B. Decentralized Repair Mechanism

Let messages be stored among n servers in a decentralized erasure code based storage system. After a period of time, some servers fail. Let the number of remaining servers be αn , where $\alpha < 1$. By the results [16], [17], [18], any k remaining servers can recover the message with probability $1 - k/p - o(1)$. To repair the system from $(1 - \alpha)n$ server failures, $(1 - \alpha)n$ new servers join the system. We shall call a remaining server as an old server and a newly joining one as a new server. A repair procedure is initiated by new servers (see Fig. 2). After executing the repair procedure, the storage system is recovered from server failures so that any k servers, no matter new or old ones, shall recover the message with an overwhelming probability.

Repair procedure. New server SS_j performs the following steps:

- 1) Query u randomly chosen old servers, $\text{SS}_{j_1}, \text{SS}_{j_2}, \dots, \text{SS}_{j_u}$. A queried old server SS_{j_i} returns the stored codeword symbol and coefficients $(c_{j_i}, g_{j_i,1}, g_{j_i,2}, \dots, g_{j_i,k})$.
- 2) Choose a random coefficient z_{j_i} for a received $(c_{j_i}, g_{j_i,1}, g_{j_i,2}, \dots, g_{j_i,k})$.
- 3) Encode all received data into a new codeword symbol and the corresponding coefficients $(\tilde{c}_j, \tilde{g}_{j,1}, \tilde{g}_{j,2}, \dots, \tilde{g}_{j,k})$:

$$\tilde{c}_j = \sum_{1 \leq i \leq u} z_{j_i} c_{j_i}, \quad \tilde{g}_{j,s} = \sum_{1 \leq i \leq u} z_{j_i} g_{j_i,s}, \quad 1 \leq s \leq k$$

- 4) Store the resulting $(\tilde{c}_j, \tilde{g}_{j,1}, \tilde{g}_{j,2}, \dots, \tilde{g}_{j,k})$.

By considering communication cost of establishing network connections between servers, we want a smaller u . A larger u means that the new server queries more codeword symbols from old servers. The combination of these queried codeword symbols contains more information about the message. Therefore, we need to carefully select u . Apparently, if $u \geq k$, more than k codeword symbols are queried and they are sufficient to recover the message with an overwhelming probability. The combination of these codeword symbols in the new server, together with the codeword symbols from

other $k - 1$ servers, should provide enough information to recover the message. On the other hand, if $u < k$, the queried codeword symbols are not sufficient to recover the message and their combination contains less information about the message. We are interested in finding out how smaller u can be such that the combination of the queried codeword symbols still provides sufficient information, when together with other codeword symbols, to recover the message with an overwhelming probability.

C. Main Result

We assume that $n = ak^c$ and $\alpha n = k^d$ for some constant a, c, α , and d , where $c \geq 1$, $\alpha < 1$, and $d > 1$. This assumption can be generally applied to decentralized erasure code based storage systems. Our results are given in Theorem 1 and Theorem 2. Proofs are provided in subsequent subsections.

Theorem 1 shows that in a decentralized erasure code based storage system with n servers, our repair mechanism with $u = k$ recovers the system from $(1 - \alpha)n$ server failures.

Theorem 1. *Let $n = ak^c$ for some constants a and c , where $c \geq 1$. Let the number αn of old servers be k^d , where $\alpha < 1$ and $d > 1$. Let the system be repaired by our repair mechanism with $u = k$. Consider the event of a successful data retrieval that k randomly chosen servers from new and old servers recover a message. The probability of a successful retrieval is at least $1 - \frac{2k}{p} - o(1)$.*

Theorem 2 shows the bound on u for our repair mechanism. The bound reveals the opportunities for $u < k$.

Theorem 2. *Let $n = ak^c$ and $\alpha n = k^d$ for some constants a, c, α , and d , where $c \geq 1$, $\alpha < 1$, and $d > 1$. Let the parameter u be set such that*

$$u \geq \min\left\{k, \max\left\{\frac{2k}{(d-1)\ln k}, \left(\frac{k}{(d-1)\ln k} + \frac{d}{d-1}\right)\right\}\right\}$$

After the system is repaired by our repair mechanism, the probability of a successful retrieval is at least $1 - \frac{2k}{p} - o(1)$.

Corollary 1. *When $d > \frac{k}{\ln k}$, it is sufficient to have $u \geq \min\left\{k, \frac{k}{(d-1)\ln k} + \frac{d}{d-1}\right\}$. When $d \leq \frac{k}{\ln k}$, it is sufficient to have $u \geq \min\left\{k, \frac{2k}{(d-1)\ln k}\right\}$.*

Proof: When $d > \frac{k}{\ln k}$, we have $\frac{2k}{(d-1)\ln k} < \left(\frac{k}{(d-1)\ln k} + \frac{d}{d-1}\right)$. When $d \leq \frac{k}{\ln k}$, we have $\frac{2k}{(d-1)\ln k} \geq \left(\frac{k}{(d-1)\ln k} + \frac{d}{d-1}\right)$. ■

From Theorem 2, with a fixed d , u can be less than k when k is sufficiently large. Similarly, with a fixed k , u can be less than k when d is sufficiently large. It implies that when available servers are abundant, a new server can query fewer servers for replenishing a codeword symbol.

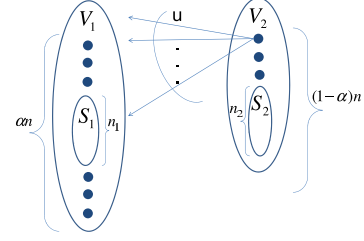


Figure 3. The random bipartite graph \mathbb{G} of the repair mechanism.

D. Proof of Theorem 1

Let E_0 be the event that k servers randomly chosen from αn old servers recover a message. Our assumption on αn old servers is that $\Pr[E_0] \geq 1 - k/p - o(1)$. Let n_1 and n_2 be the numbers of queried old servers and queried new servers, respectively. Thus, $n_1 + n_2 = k$. Let the event E_1 be that k servers randomly chosen from old and new servers recover a message. Our goal is to show that $\Pr[E_1] \geq 1 - 2k/p - o(1)$. We divide the event E_1 into subevents as shown in Equation (1).

$$\begin{aligned} \Pr[E_1] &= \Pr[E_1 | n_1 = k] \Pr[n_1 = k] \\ &\quad + \Pr[E_1 | n_1 < k] \Pr[n_1 < k] \end{aligned} \quad (1)$$

When $n_1 = k$, we directly obtain:

$$\Pr[E_1 | n_1 = k] = \Pr[E_0] \geq 1 - k/p - o(1) > 1 - 2k/p - o(1)$$

When $n_1 < k$, we model the repair mechanism as a random bipartite graph \mathbb{G} and analyze the random graph.

Illustrated in Fig. 3, the random bipartite graph is $\mathbb{G} = (V_1, V_2, E)$, where V_1 and V_2 are vertex sets with $|V_1| = \alpha n$ and $|V_2| = (1 - \alpha)n$ and E is the edge set. Each vertex v_i in V_1 represents an old server SS_i and each vertex v_j in V_2 represents a new server SS_j . There is an edge (v_i, v_j) between vertices $v_i \in V_1$ and $v_j \in V_2$ if and only if the new server SS_j queries the old server SS_i . Note that a new server queries k old servers. A set S of k servers represents a set of servers chosen for data retrieval. The set S consists of two subsets S_1 and S_2 of vertices in \mathbb{G} , where $S_1 \subseteq V_1$ with $|S_1| = n_1$ and $S_2 \subseteq V_2$ with $|S_2| = n_2$. Event E_2 is that there is a maximal matching from S_2 to $V_1 \setminus S_1$. We divide the event E_1 conditioned on $n_1 < k$ into subevents as shown in Equation (2), where \bar{E}_2 is the complement event of E_2 .

$$\begin{aligned} &\Pr[E_1 | n_1 < k] \Pr[n_1 < k] \\ &= \Pr[E_1 | E_2 \wedge (n_1 < k)] \Pr[E_2 | n_1 < k] \Pr[n_1 < k] \\ &\quad + \Pr[E_1 | \bar{E}_2 \wedge (n_1 < k)] \Pr[\bar{E}_2 | n_1 < k] \Pr[n_1 < k] \end{aligned} \quad (2)$$

We need Lemma 1 and Lemma 3 to formulate relations between events E_1 and E_2 to complete this proof.

Lemma 1. $\Pr[E_1 | E_2 \wedge (n_1 < k)] \geq 1 - 2k/p - o(1)$

Proof: Let $N(S_2) \subseteq V_1$ be the set of neighbors of S_2 .

When E_2 happens, there is a maximal matching from S_2 to $V_1 \setminus S_1$. That is, a subset $S'_2 \subseteq N(S_2) \setminus S_1$ exists with $|S'_2| = n_2$

Let K be the $k \times k$ matrix formed by coefficients from queried servers in $S_1 \cup S_2$. When K is invertible, E_1 happens. Let K_1 be the $k \times k$ matrix formed by coefficients from the servers in $S_1 \cup S'_2$. Since $S_1 \cup S'_2$ is a subset of k vertices in V_1 , K_1 is invertible with probability at least $1 - k/p - o(1)$. Since the subgraph induced by S_2 and S'_2 has a perfect matching, K has full rank if K_1 has full rank. Moreover, each row in K can be expressed as a linear combination of rows in K_1 . Thus, K can be expressed as $T \times K_1$ for some $k \times k$ matrix T . Entries of T are randomly and independently determined by new servers. To have K invertible, K_1 and T must be invertible. When K_1 is invertible, T is invertible with probability at least $1 - k/p$ according to the Schwartz-Zippel Theorem. Thus, we have

$$\begin{aligned} & \Pr[E_1 | E_2 \wedge (n_1 < k)] \\ &= \Pr[K \text{ is invertible} | E_2 \wedge (n_1 < k)] \\ &\geq \Pr[K_1 \text{ is invertible} \wedge T \text{ is invertible} | E_2 \wedge (n_1 < k)] \\ &\geq (1 - k/p - o(1)) \times (1 - k/p) \\ &\geq 1 - 2k/p - o(1) \end{aligned}$$

■

Lemma 2. (Hall's Theorem) *If and only if for any subset $B \subseteq S_2$, the number of neighbors of B in $V_1 \setminus S_1$ is no less than the size of B , i.e., $|N(B) \setminus S_1| \geq |B|$, where $N(B) \subseteq V_1$ is the set of neighbors of B , there exists a maximal matching from S_2 to $V_1 \setminus S_1$.*

Lemma 3. $\Pr[E_2 | n_1 < k] = 1$

Proof: When $u = k$, each vertex v in S_2 has k neighbors in V_1 . For all possible B , where $1 \leq |B| \leq n_2$,

$$|N(B) \setminus S_1| \geq k - n_1 = n_2 \geq |B|.$$

Hence, $\Pr[E_2 | n_1 < k] = 1$. ■

From Equation (1), Lemma 1, and Lemma 3, we have

$$\begin{aligned} \Pr[E_1] &= \Pr[E_1 | n_1 = k] \Pr[n_1 = k] + \Pr[E_1 | n_1 < k] \Pr[n_1 < k] \\ &\geq \Pr[E_1 | n_1 = k] \Pr[n_1 = k] \\ &\quad + \Pr[E_1 | E_2 \wedge (n_1 < k)] \Pr[E_2 | n_1 < k] \Pr[n_1 < k] \\ &\geq (1 - k/p - o(1)) \Pr[n_1 = k] \\ &\quad + (1 - 2k/p - o(1)) \Pr[n_1 < k] \\ &\geq 1 - 2k/p - o(1) \end{aligned}$$

It concludes this proof.

E. Proof of Theorem 2

The proof of Theorem 2 is similar to the proof of Theorem 1 except for the analysis of the random graph. To ease the analysis, the original repair procedure is modified to that a new server randomly queries an old server u

times with replacement. Thus, a new server may query less than u distinct old servers. The modification leads to a different random graph. The probability of a maximum matching from S_2 to $V_1 \setminus S_1$ in the new random graph is smaller than that in the original random graph. Hence the probability in the original random graph is underestimated. Let $\mathbb{G}' = (V_1, V_2, E')$ be the random bipartite graph, where $|V_1| = \alpha n$, $|V_2| = (1 - \alpha)n$, and E' is the edge set. Let event E'_2 is that there is a maximal matching from S_2 to $V_1 \setminus S_1$. Again, we need Lemma 1 and Lemma 4 for relations between events E_1 and E'_2 to complete this proof.

Lemma 4. $\Pr[E'_2 | n_1 < k] \geq 1 - o(1)$

Proof: We use Lemma 2 (Hall's theorem) and Lemma 5 to bound the probability $\Pr[E'_2 | n_1 < k]$. Lemma 5 is a bound for C_y^x (Due to limited space, the proof for Lemma 5 is omitted):

Lemma 5. $C_y^x \leq \left(\frac{x(x-y+1)}{y}\right)^{\frac{y}{x}}$

When there exists a subset $B \subseteq S_2$ where $|N(B) \setminus S_1| < |B|$, no maximal matching from S_2 to $V_1 \setminus S_1$ exists. We consider every possible subset B and overestimate the probability of the complement event of E'_2 by a union bound.

$$\begin{aligned} & \Pr[\exists B \subseteq S_2, |N(B) \setminus S_1| < |B|] \\ &\leq 2^k \cdot \max_{B \subseteq S_2} \{\Pr[|N(B) \setminus S_1| < |B|]\} \end{aligned}$$

Let $|B| = t$, where $1 \leq t \leq n_2$. The event that some subset B exists for $|N(B) \setminus S_1| < |B|$ is equivalent to the event that some subset A exists where $A \subseteq V_1 \setminus S_1$, $|A| \leq t - 1$, and $A \cup S_1 \supseteq N(B)$

$$\begin{aligned} & \Pr[|N(B) \setminus S_1| \leq |B|] \\ &= \Pr[\exists A, |A| \leq t - 1, A \cup S_1 \supseteq N(B)] \\ &\leq C_{t-1}^{\alpha n - n_1} \left(\frac{k-1}{\alpha n}\right)^{tu} \quad (\text{Lemma 5}) \\ &\leq \left(\frac{2(\alpha n - n_1)(\alpha n - n_1 - t + 2)}{t}\right)^{\frac{t-1}{2}} \left(\frac{k}{\alpha n}\right)^{tu} \end{aligned}$$

Since we want $\Pr[\exists B \subseteq S_2, |N(B) \setminus S_1| < |B|] < e^{-k}$, it is sufficient to have:

$$\left(\frac{2(\alpha n - n_1)(\alpha n - n_1 - t + 2)}{t}\right)^{\frac{t-1}{2}} \left(\frac{k}{\alpha n}\right)^{tu} < e^{-2k} \quad (3)$$

Now we substitute $\alpha n = k^d$ in Equation (3) and overestimate the left hand side:

$$\left(\frac{2k^{2d}}{t}\right)^{\frac{t-1}{2}} k^{(1-d)tu} < e^{-2k} \quad (4)$$

We take nature logarithm on both sides of Equation (4) and

obtain the bound on u :

$$u > \frac{(t-1)(\ln 2 + 2d \ln k - \ln t) + 4k}{2(d-1)t \ln k}$$

When $t = 1$, the bound becomes $\frac{2k}{(d-1)\ln k}$. When $2 \leq t \leq k$, it is sufficient to have $u > \frac{d}{d-1} + \frac{k}{(d-1)\ln k}$. Combining the result from Theorem 1, we obtain the requirement on u :

$$u \geq \min\{k, \max\{\frac{2k}{(d-1)\ln k}, \left(\frac{k}{(d-1)\ln k} + \frac{d}{d-1}\right)\}\}.$$

When u meets this requirement, $\Pr[E'_2 | n_1 < k] \geq 1 - e^{-k} = 1 - o(1)$. ■

From Equation (1), Lemma 1, and Lemma 4, we have

$$\begin{aligned} & \Pr[E_1] \\ &= \Pr[E_1 | n_1 = k] \Pr[n_1 = k] + \Pr[E_1 | n_1 < k] \Pr[n_1 < k] \\ &\geq \Pr[E_1 | n_1 = k] \Pr[n_1 = k] \\ &\quad + \Pr[E_1 | E'_2 \wedge (n_1 < k)] \Pr[E'_2 | n_1 < k] \Pr[n_1 < k] \\ &\geq 1 - 2k/p - o(1) \end{aligned}$$

It concludes this proof.

IV. NUMERICAL ANALYSIS AND PARAMETERIZED COMPARISON

We conducted a numerical analysis by using traces of several real systems. We also compare our decentralized repair mechanism with other robustness management mechanisms.

A. Numerical Analysis

We introduce two key parameters from real systems. One is the number n of servers. The other is the fraction f of failed servers per day. From traces of real systems, the number of servers varies as well as the fraction f over time. We bring the average values into our repair mechanism in the theoretical setting.

Traces. We quote statistics from [2] by Dimakis et al. The statistics summarized parameters from traces of 4 real systems: desktop PCs within Microsoft Corporation [19], Gnutella peers [20], Skype superpeers [21], and the PlanetLab. The average number n of servers and the average fraction f of failed servers per day are shown in Table I.

The parameter u represents the communication cost and only depends on k and d . We are interested in the value of u with different system scales n and different numbers k^d of available servers. In a lazy strategy for repairing a system, the number k^d determines a threshold value that triggers execution of a repair procedure. From Theorem 2 and Corollary 1, we illustrate the numerical results in Table II. With a fixed k , when d gets larger, u can be smaller. With a fixed d , when k gets larger, u is much smaller than k . It shows that when remaining servers are abundant, the robustness maintaining cost is lower. More importantly, the number of servers queried by a new server can be smaller

than k . For example, when $k = 8$ and $n = 4096$ servers are available, u can be set to only 3.

Survival duration. Since our repair mechanism recovers the storage system from multiple server failures, a strategy for periodical repairing is supported. We are interested in the duration time that a storage system can stand against server failures without any repairing. That is, the system still have sufficient servers to perform the repair procedure when needed. This period of time is called survival duration. We consider various αn remaining servers. We bring the fraction f of failed servers per day into the scenario. With a fixed f , the system losses nf servers per day if no repair procedure is performed. The survival duration in days is estimated as $\lfloor (n - \alpha n) / \lceil nf \rceil \rfloor$. When $n \gg \alpha n$, the survival duration is close to $1/f$. We choose u as small as possible under the limitation that $\alpha n < n$. The numerical results are given in Table III. For example, in the case of PlanetLab, the system has 303 servers and 0.017% of servers fail per day on average. When $k = 4$, we set $u = 3$, which is the smallest one with $\alpha n < n$ (see Table II). The threshold value of available servers is 16. Thus, the system stands against server failures for 47 days. After the 47th day, the system would not have sufficient servers for the repair procedure to work.

B. Parameterized Comparison

As introduced in Section II, some repair mechanisms can be applied to decentralized erasure code based storage systems. From the family of regenerating codes [2], we choose two mechanisms, the minimum bandwidth regime (MBR) and the minimum storage regime (MSR). The two mechanisms result in two extreme points on the trade off curve. MBR minimizes the repair bandwidth and MSR minimizes the storage cost. We also compare our mechanism with the mutual cooperative recovery (MCR) mechanism [10] and self-repair homomorphic codes (SRHC) [11] since they both consider multiple server failures.

Let l be the size of a message in bits. We compare our mechanism with them in the following items: 1) the number u of required connections per server failure, 2) the number of repaired server failures, 3) required bandwidth for replenishing a codeword symbol in bits, 4) storage cost per server in bits, and 5) method type. The 5th item is an indicator of whether the mechanism is suitable in a decentralized environment. When the repair procedure is independent of missing codeword symbols, we call such mechanism "symmetric". In other words, an asymmetric repair mechanism uses different steps for different patterns of missing codeword symbols. For example, SRHC is asymmetric since it regenerates a codeword symbol from a specific set of survival codeword symbols. The comparison is summarized in Table IV.

Regenerating codes show that repair bandwidth can be less than the size l of the message when a new server

Trace	Microsoft PCs	Gnutella	Skype	PlanetLab
n : average number of nodes	41970	1846	710	303
f : fraction of failed node per day	0.038	0.3	0.12	0.017

Table I
STATISTICS OF SYSTEM TRACES [2].

$k = 4$					$k = 8$					$k = 16$					
d	2	3	4	5	6	d	2	3	4	5	d	2	3	4	5
u	3	3	3	3	2	u	6	4	3	3	u	8	5	4	3
k^d	16	64	256	1024	4096	k^d	64	512	4096	32768	k^d	256	4096	65536	1048576

Table II
NUMERICAL ANALYSIS FOR THE NUMBER u FOR DIFFERENT k AND αn .

Trace	Microsoft		Gnutella		Skype		PlanetLab	
n	41970		1846		710		303	
f	0.038		0.3		0.12		0.017	
k	4	8	4	8	4	8	4	8
u	3	3	3	4	3	4	3	6
αn	16	4096	16	512	16	512	16	64
Survival duration (days)	26	23	3	3	8	2	47	39

Table III
NUMERICAL ANALYSIS FOR SURVIVAL DURATION IN DAYS.

	u	server failures	bandwidth	storage	type
MBR [2]	$n - 1$	single	$\frac{(2n-2)l}{(2n-k-1)k}$	$\frac{(2n-2)l}{(2n-k-1)k}$	symmetric
MSR [2]	$k + 1$	single	$\frac{(n-1)l}{(n-k)k}$	$\frac{l}{k}$	symmetric
MCR [10]	$n - 1$	multiple	$\frac{(n-1)l}{(n-k)k}$	$\frac{l}{k}$	symmetric
SRHC [11]	$< k$	multiple	$\frac{ul}{k}$	$\frac{l}{k}$	asymmetric
Our work	$< k$	multiple	$\frac{ul}{k}$	$\frac{l}{k}$	symmetric

Table IV
COMPARISON OVER REPAIR MECHANISMS.

queries more than k servers. However, they only tolerate one server failure. MCR tolerates multiple server failures, but the number of required connections for repairing a failure is $n - 1$. In other words, a new server has to communicate with all other servers in the storage system. SRHC is a novel way to recover the system from multiple server failures with $u < k$. But, SRHC is not suitable for distributed or decentralized environment because it is asymmetric.

Our mechanism outperforms existing ones in terms of the communication cost under the same consideration of efficiency cost for storage. A new server queries less than k servers and the required bandwidth is less than l . At the same time, the storage cost is as less as the cost of the MSR. Moreover, our repair mechanism recovers a decentralized erasure code based storage system from multiple server failures.

The sacrifice is the probability of a successful data retrieval. The probabilities of a successful data retrieval in MBR, MSR, and MCR are all 1's. Since SRHC exactly

regenerates missing codeword symbols, the probability is 1 as well. While our mechanism has lower communication cost, the probability of a successful data retrieval is $1 - 2k/p - o(1)$. However, by choosing a sufficient large p , the probability $1 - 2k/p - o(1)$ is overwhelming. Moreover, the probability can be dramatically increased by letting a user query more than k servers for data retrieval.

V. CONCLUSION AND FUTURE WORK

We consider the measurement of communication cost in terms of the number u of connections that a new server has to establish. Our repair mechanism provides flexible adjustment between u and the number of remaining servers. More importantly, our results confirm that to repair a server failure, a new server can query less than k servers.

Our repair mechanism symmetrically repairs multiple server failures of decentralized erasure code based storage systems. Thus, a lazy repair strategy or a periodical repair strategy can be taken upon our repair mechanism. It is

compatible with most decentralized erasure code based storage systems without any change in encoding and decoding methods. Both theoretical and numerical results show that our decentralized repair mechanism is efficient and practical.


In our repair mechanism, new servers do not communicate with each other during the repair procedure. In some practical cases, they can exchange information for repairing. Intuitively, mutual communications among new servers can further decrease the number u . Exploring the quantity of possible improvement is our work in progress. Statistical simulation results are also required to demonstrate the practicality of our repair mechanism.

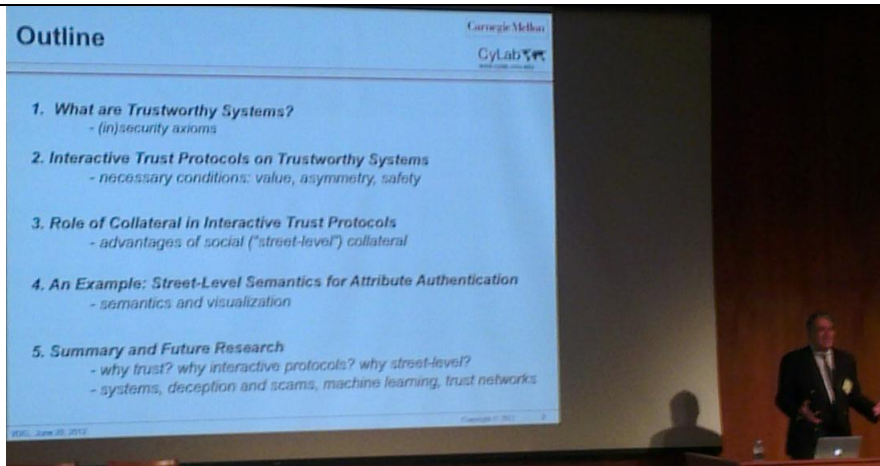
ACKNOWLEDGMENT

The research was supported in part by projects ICTL-100-Q707, ATU-100-W958, NSC 98-2221-E-009-068-MY3, NSC 100-2218-E-009-003-, and NSC 100-2218-E-009-006-.

REFERENCES

- [1] Rodrigo Rodrigues and Barbara Liskov. High availability in dhts: Erasure coding vs. replication. In *Proceedings of the 4th International Workshop on Peer-to-Peer Systems - IPTPS 2005*, 2005.
- [2] Alexandros G. Dimakis, Brighten Godfrey, Martin J. Wainwright, and Kannan Ramchandran. Network coding for distributed storage systems. In *Proceedings of the 26th IEEE International Conference on Computer Communications – INFOCOM 2007*, pages 2000–2008. IEEE, 2007.
- [3] Y. Wu, A. G. Dimakis, and K. R. Ramchandran. Deterministic regenerating codes for distributed storage systems. In *Proceedings of the 45th annual Allerton conference on Communication, control, and computing*, Allerton’07, pages 1243–1249. IEEE Press, 2007.
- [4] Alexandros G. Dimakis, Brighten Godfrey, Yunnan Wu, Martin J. Wainwright, and Kannan Ramchandran. Network coding for distributed storage systems. *IEEE Transactions on Information Theory*, 56(9):4539–4551, 2010.
- [5] K. V. Rashmi, Nihar B. Shah, P. Vijay Kumar, and Kannan Ramchandran. Explicit construction of optimal exact regenerating codes for distributed storage. In *Proceedings of the 47th annual Allerton conference on Communication, control, and computing*, Allerton’09, pages 1243–1249. IEEE Press, 2009.
- [6] Nihar B. Shah, K. V. Rashmi, and P. Vijay Kumar. A flexible class of regenerating codes for distributed storage. In *Proceedings of IEEE symposium on Information Theory 2010*, pages 1943–1947. IEEE Press, 2010.
- [7] Soroush Akhlaghi, Abbas Kiani, and Mohammad Reza Ghanavati. A fundamental trade-off between the download cost and repair bandwidth in distributed storage systems. In *Proceedings of IEEE International Symposium on Network Coding 2010 – NetCod*, pages 1–6, 2010.
- [8] Salim El Rouayheb and Kannan Ramchandran. Fractional repetition codes for repair in distributed storage systems. In *Proceedings of the 48th annual Allerton conference on Communication, control, and computing*, Allerton’10. IEEE Press, 2010.
- [9] Alessandro Duminuco and Ernst W. Biersack. Hierarchical codes: A flexible trade-off for erasure codes in peer-to-peer storage systems. *Peer-to-Peer Networking and Applications*, 3(1):52–66, 2010.
- [10] Yuchong Hu, Yinlong Xu, Xiaozhao Wang, Cheng Zhan, and Pei Li. Cooperative recovery of distributed storage systems from multiple losses with network coding. *Selected Areas in Communications, IEEE Journal on*, 28(2):268–276, 2010.
- [11] Frederique Oggier and Anwitaman Datta. Self-repairing homomorphic codes for distributed storage systems. In *Proceedings of the 30th IEEE international conference on Computer communications 2011*. IEEE Press, 2011.
- [12] Theodoros K. Dikaliotis, Alexandros G. Dimakis, and Tracey Ho. Security in distributed storage systems by communicating a logarithmic number of bits. In *Proceedings of IEEE symposium on information theory 2010*. IEEE Press, 2010.
- [13] K. V. Rashmi, Nihar B. Shah, and P. Vijay Jumar. Enabling node repair in any erasure code for distributed storage, 2011.
- [14] Sameer Pawar, Salim El Rouayheb, and Kannan Ramchandran. On secure distributed data storage under repair dynamics. Technical Report UCB/EECS-2010-18, University of California Berkeley, EECS, 2010.
- [15] Dimitris S. Papailiopoulos and Alexandros G. Dimakis. Distributed storage codes meet multiple-access wiretap channels. In *Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing*, Allerton’10, pages 1420–1427, 2010.
- [16] Alexandros G. Dimakis, Vinod Prabhakaran, and Kannan Ramchandran. Decentralized erasure codes for distributed networked storage. *IEEE/ACM Transactions on Networking*, 14:2809–2816, 2006.
- [17] Hsiao-Ying Lin and Wen-Guey Tzeng. A secure decentralized erasure code for distributed network storage. *IEEE transactions on Parallel and Distributed Systems*, 21:1586–1594, 2010.
- [18] Hsiao-Ying Lin and Wen-Guey Tzeng. A secure erasure code based cloud storage system with secure data forwarding. manuscript.
- [19] William J. Bolosky, John R. Douceur, David Ely, and Marvin Theimer. Feasibility of a serverless distributed file system deployed on an existing set of desktop pcs. *ACM SIGMETRICS Performance Evaluation Review*, 28:34–43, 2000.
- [20] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Computing and Networking*, 2002.
- [21] Saikat Guha, Neil Daswani, and Ravi Jain. An experimental study of the skype peer-to-peer voip system. In *Proceedings of the 5th International Workshop on Peer-to-Peer Systems*, 2006.

報告人姓名	曾文貴	服務機構 及職稱	交通大學資工系 教授
時間 會議 地點	101 年 6 月 20 日至 101 年 6 月 22 日 (出國時間為 101 年 6 月 18 日至 101 年 6 月 24 日) 美國華盛頓特區 NIST Administrator Building 101		
會議 名稱	2012 IEEE International Conference on Software Security and Reliability (SERE 2012)		
出國目的/ 發表論文題 目	發表論文： 論文作者於題目：Hsiao-Ying Lin, John Kubiawicz and Wen-Guey Tzeng. A Secure Fine-Grained Access Control Mechanism for Networked Storage System. <i>In the Sixth IEEE International Conference on Software Security and Reliability (IEEE SERE 2012)</i> , June 2012.		
內容包括下列各項：			
一、 參加會議經過(含照片)			
<p>本人於 18 日從台灣搭機，當日抵達，19 日調整時差，20 開始參加會議，會議舉辦期間為 6 月 20 日至 6 月 22 日，參加完會議後，於 23 日離開華盛頓特區，24 日到達台灣。於會議舉辦期間，本人參加會議行程，詳細行程資訊按時間順序整理如下：</p> <ul style="list-style-type: none"> ● 6 月 20 日： 會議首日，由謝續平教授協助聯繫當地的同學來接我們一行三人抵達 NIST 的 101 大樓。進入 NIST 區域需要持有一份通行文件與一份含照片的個人識別證件，經過警察核對之後才能進入，門禁相當森嚴。我們抵達會場時約為早上 9 點半。會議地點在 NIST 區域的 A101 大樓。 			
			
<p>報到時，拿到大會時程，注意到自己需要在 22 日下午主持一場議程(session)。</p> <p>第一場 Keynote speech 是由 Virgil Gligor 主講。</p>			



會議除了 keynote speech 之外的議程都有平行議程，同一時間有三個議程進行。參加的議程內容如下所示：

- 10:30~12:00
Room 3
- Session 1C: IA Workshop I**
- *Towards a Model Based Security Testing Approach of Cloud Computing Environments*
Philipp Zech¹, Michael Felderer¹, and Ruth Breu²
¹University of Innsbruck, Austria
²Research Group Quality Engineering, Austria
 - *Designing System Security with UML Misuse Deployment Diagrams*
Susan Lincke, Timothy Knautz, and Misty Lowery
University of Wisconsin–Parkside, USA
 - *A Proposal to Prevent Click–Fraud Using Clickable CAPTCHAs*
Rodrigo Alves Costa, Ruy J. G. B. de Queiroz, and Elmano Ramalho Cavalcanti
University Federal de Campina Grande, Brazil

下午的議程部分，首先進行的是第二場的 keynote speech。

Invited Talk

Philip Laplante – Safe and Secure Software Systems and the Role Professional Licensure

下午參加的議程為：

- 14:00~15:00
Room 3
- Session 2C: IA Workshop II**
- *Comparing Static Security Analysis Tools Using Open Source Software*
Ryan McLean
Air Force Institute of Technology, USA
 - *Undesirable Aspect Interactions: a Prevention Policy for Three Aspect Fault Types*
Arsène Sabas¹, Subash Shankar², Virginie Wiels³, and Michel Boyer¹
¹Université de Montréal, Canada
²City University of New York, USA
³Onera – the French Aerospace Lab, France

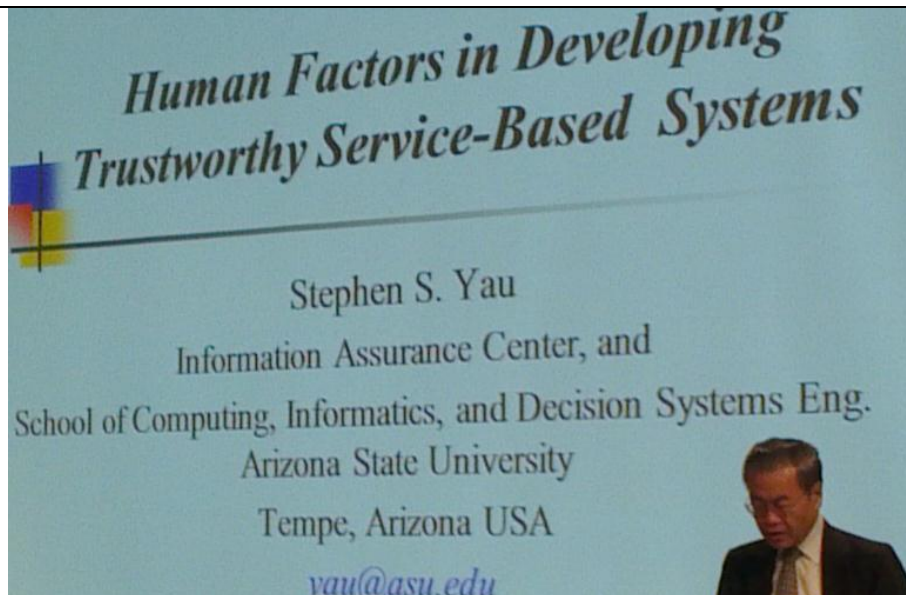
Session 3A: Quality Analysis

- *An Autonomic Framework for Integrating Security and Quality of Service Support in Databases*
Firas Alomari and Daniel Menasce
George Mason University, USA
- *VRank: A Context–Aware Approach to Vulnerability Scoring and Ranking in SOA*
Jianchun Jiang¹, Liping Ding¹, Ennan Zhai¹, and Ting Yu²
¹Chinese Academy of Sciences, China
²North Carolina State University, USA
- *Security Impacts of Virtualization on a Network Testbed*
Yu Lun Huang, Bortong Chen, Ming Wei Shih, and Chien Yu Lai
National Chiao Tung University, Taiwan

這天大會會提供一些小點心，隨後我們便搭乘旅館的接駁車回旅館休息，並且在旅館附近用晚餐。

● 6月21日：

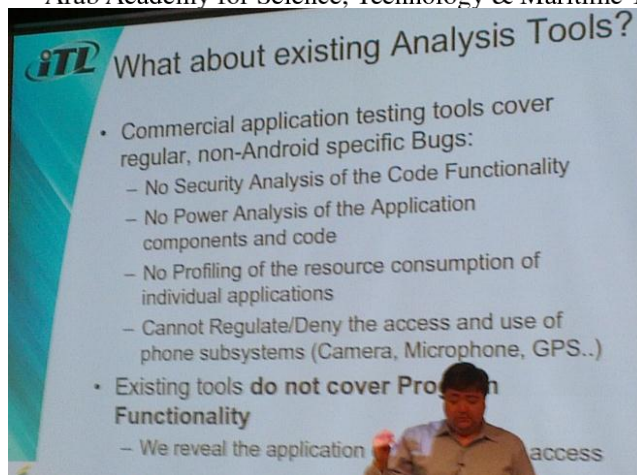
會議第二天，是由第三場 Keynote speech 開始。



早上參加的議程為：

Session 4C: IA Workshop IV

- *Scalable Software Testing for Android: Challenges & Opportunities (Invited Talk)*
Angelos Stavrou
George Mason University, USA
- *Secure PC Platform Based on Dual-Bus Architecture*
Hesham El Zouka
Arab Academy for Science, Technology & Maritime Transport, Egypt



下午則是參加：

Session 5C: IA Workshop V

- *A Privacy Preserving Smart Metering System Supporting Multiple Time Granularities*
Hsiao Ying Lin, Shiu-an Tzuo Shen, and Bao-Shuh P. Lin
National Chiao Tung University, Taiwan
- *An Investigation of Classification-Based Algorithms for Modified Condition/ Decision Coverage Criteria*
Jun-Ru Chang¹, Chin-Yu Huang², and Po-Hsi Li²
¹Realtek Semiconductor Corporation, Taiwan
²National Tsing Hua University, Taiwan

下午的活動含有一個自助參訪與晚宴。兩項活動都是在 NIST 101 大樓內舉行，我們在參訪活動中，找到了牛頓的蘋果樹的後代，以及參訪了 NIST 的博物館 (Museum)：



NEWTON APPLE TREE

"SCIENCE HAS ITS TRADITIONS AS WELL AS ITS FRONTIERS"

THIS TREE IS A DIRECT DESCENDANT OF THE ORIGINAL TREE WHOSE FRUIT GAVE INSPIRATIONAL IMPETUS TO ISAAC NEWTON'S THEORY OF GRAVITATIONAL FORCES. IT WAS NURTURED BY THE U.S. DEPARTMENT OF AGRICULTURE AND TRANSPLANTED HERE ON THE GROUNDS OF THE NATIONAL BUREAU OF STANDARDS APRIL 1966.

晚上的接待晚宴於 NIST 內進行，下圖是擔任 Program Chair 的謝續平教授致詞以及用餐過程中窗外突然出現的鹿：



● 6月22日：

會議第三天，我的論文報告被安排在這天的下午第二段時間，由林孝盈博士博報告。這天亦由一個 keynote speech 開始進行會議，這天會議的上午日程如下所示：

09:00~10:00	Keynote Speech III Huimin Lin – Checking Safety Properties of Concurrent Programs
10:00~10:30	Coffee Break
10:30~12:00 Room 1	Session 7A: Access Control & Authentication <ul style="list-style-type: none"> • <i>A Secure Fine-Grained Access Control Mechanism for Networked Storage Systems</i> Hsiao Ying Lin¹, John Kubiawicz² and Wen Guey Tzeng¹ ¹National Chiao Tung University, Taiwan ²University of California Berkeley, USA • <i>Mitigating Insider Threat without Limiting the Availability in Concurrent Undeclared Tasks</i> Qussai Yaseen, and Brajendra Panda University of Arkansas, USA • <i>A New Non-Intrusive Authentication Method based on the Orientation Sensor for Smartphone Users</i> Chien-Cheng Lin¹, Chin-Chun Chang¹, Deron Liang², and Ching-Han Yang² ¹National Chiao Tung University, Taiwan ²University of California Berkeley, USA

在報告之後，有一位學者提出三個問題，分別是針對取消授權，儲存成本，以及與其他存取控制方式的比較討論。上午議程結束後，與此學者討論了在結合應用系統與密碼學工具上的經驗。

這天下午的日程如下所示，這兩個議程皆由本人擔任議程主席(Session chair):

Session 8B: Student Doctoral Program II

- *A Survey of Software Testing in the Cloud,*
Koray Inçkıl¹, İsmail Ari², and Hasan SÖzer²
¹TÜBİTAK BILGEM Information Technologies Institute, Turkey
²Özyeğin University, Turkey
- *A Novel Method for Modeling Complex Network of Software System Security*
Hailin Li, Yadi Wang, and Jihong Han
Zhengzhou University, China
- *Thinking Towards a Pattern Language for Predicate Based Encryption Crypto-Systems*
Jan de Muijnck-Hughes and Ishbel Duncan
University of St Andrews, United Kingdom

Session 9C: Fast Abstract II

- *Intelligent Biological Security Testing Agents*
Ishbel Duncan
University of St Andrews, United Kingdom
- *Attestation & Authentication for USB Communications*
Zhaohui Wang and Angelos Stavrou
George Mason University, USA
- *Analysis of Android Applications' Permissions*
Ryan Johnson¹, Zhaohui Wang¹, Corey Gagnon², and Angelos Stavrou¹
¹George Mason University, USA
²James Madison University, USA

至此，會議順利進行結束。

● 6月23日:

早上 8:30 離開飯店，搭乘地鐵前往雷根機場，在機場除了到航空櫃台報到，進行行李檢查，亦通過繁複的安全檢查，足見美國對於機場安全的謹慎。在底特律及東京轉機後，於台灣時間 6 月 24 日晚間 7 點抵達桃園機場，結束此次行程。

二、與會心得

這次與會在研究方面有多項心得，首先研究學術議題與潮流方面，目前針對軟體安全與系統安全的研究大都需要檢驗非常底層的東西，例如原始碼(source code)或執行檔(binary code)，以發掘潛在的軟體弱點或系統弱點，因此需要大量的計算，非常適合雲端的架構來執行，另一方面，利用 Model checking 的技術來檢驗各種系統的功能與安全性也受到重視。我們發表的文章是屬於系統權限的存取控制，雖然較少的會議的參者熟悉，但是在進行報告之後，許多學者積極

的回響，可見國際學者的學術研究並不設限於自己專長的領域，對於其他相關議題也多有涉獵。這點做研究的精神值得大家學習。

在研究學術活動方面，這次與會者中，來自大陸的學者很多，他們亦積極的互相討論交流，有的學者甚至並非會議報告者，亦前來共襄盛舉，我想國內的學者應該被鼓勵多參加這些國際重要研討會。

在研究學術服務方面，不論是會議主席還是議程主席，都非常熱心的招待大家，和與會者有熱烈的互動，對於將來的學術交流或研究合作有很大的幫助。本次由謝續平教授擔任議程主席之一，他積極鼓勵台灣的師生投稿，並安排大家擔任 session chair 職位，對於提升國內學者在軟體安全研究的知名度，透過謝教授的親身示範，若能在國際學術組織中擔任要職，對於提升台灣在國際學術知名度上有相當大的影響力。

三、參觀活動(無是項活動者省略)

(略)

四、建議

透過參加國際研討會活動，可以與國際上其他學者交流，特別是透過 QA 的機會，或者是茶會休息時間進行討論，是相當寶貴的經驗。非常建議國內學者多家參與，並且最好能夠投身國際學術服務活動以提升台灣的國際學術知名度。

五、攜回資料名稱及內容

紙本議程一本，論文光碟兩片(論文集)，名牌。

其他活動照片

會議大樓門口眾與會者合影

