ISSN:0975-9646

TD-ZRP: Trigger Driven Zero Remnance Proof Technique for Self Data Removal in Web Services

Pooja Bhavsar[1], Rakesh kumar Lodhi[2]

PG Student[1], Associate Professor[2],
[1,2]Department of Computer Science & Engineering,Patel College of Science and Technology(PCST)
Bhopal, India

Abstract- **Cyber forensic is a field which applies a detailed study on information exchanges such as data, medium, devices, locations, objectives, authenticity against the defined boundaries Here, the forensic domain derives a relationship for getting the proof of information exchange between various devices and users which helps in debugging the security flaws. One of its well known phenomenon's is zero remnance proof which holds the operations for removing the residuals or intermediate data which is generated as temporary data for completing the operations.. Mainly the unauthorized access occurs from the temporary copy of data which is left mistakenly on machine. The traditional resource management system was not aware about the total temporary copies of the data and hence the cleaning operation involved there will only destruct the limited local copies. Rest of the temporary copies left somewhere on the system or network will leads to the malicious activities. This work covers the complete problem as define earlier using its TD-ZRP solution. It is a practical implementation of trigger driven zero remnance proof for self data destruction or removal and a secure data exchange using RSA public key cryptosystem. It achieves secure, robust and high performance data transition. The solution works towards getting rid of storage problem, security, untrusted source based data access and so many other problems.**

Keywords: **Software System, Cyber Forensic, Information Exchanges, Security, Self Data Removal (SDR), Lifecycle, RSA;**

I. INTRODUCTION

Web computing, information extraction and transformation services exponentially grow with increase in digital users. Here the users and the computation mechanism are continuously exchanging the information and their intermediate results. As the data is moved between various sources and machines for applying the operations of analytics on it, there is somewhere probability of security breaches involves with the process. Many organizations are collaborating for reducing the associated security risks and handle the data privacy. It prevents the unauthenticated or malicious users against the data view of any temporary instances used for the process which was mistakenly left unprotected. Thus probability of attack on this unrevoked data from the untrusted sources will be very high. Thus, the security firms or provides always have to look up for such cases using recently innovated security primitives. One of such approach is zero remnance proof (ZRP) which assures the complete removal of all the temporary copies of the data after its operations are over. Data security is one of the major area or work for the researchers for achieving a trusted computation and information exchanges. There are some concerns which need to be kept in control. These are:

- ➢ Data Privacy
- ➢ Data Owners Privacy
- ➢ Data Retriever Privacy

There is another view of the problem associated with storage optimization. The information which is generated will have certain associated values with it, which continuously changes over time. This change defines the use and its asset importance along with other security controls [1]. The phases which deal with the different phases of changing information values are covered with Information Lifecycle Management (ILM). It is the one of the core management policies involve in integrating security with zero remnance proof. The phases of ILM are create, process, migrate, archive and dispose. These phases show the changing storage media along with storage. But there is some questions which remains unanswered like at the point when the data is no longer obliged, whether it has been completely destroyed. Because of the physical properties of storage means, the data erased may be restored. This may bring about incidentally uncover of touchy information. With this work the zero remnance proof (ZRP) is used collaboratively with the information life cycle (ILM). It reduces the probability of fabricating the information by malicious user against the unprotected, temporary and changing value based data using automatic destruction.

For getting the more robust security control some of the designers include confidentiality using cryptosystems in above process. It further increases the protection of the intermediate or change value based data. As the environment is dynamic which includes lot of data transmission, storage and handling, there additionally needs to consider processing speed and computational throughput. Automative lifecycle and zero remnance based data removal require complete erasing of all the local and permanent copies after a fixed time interval of after its changing values. If the deletion was not complete and some residues metadata remains at the location of the files, then recreation of data can be performed. If the system is unable to achieve this behavior then there is left a trapdoor for attackers to renew the copies of original and fake some other services by the same. The deletion of data is quite a complex task before which the entire number of copies which is generated has to be recognized. Whenever a file is replicated some information needs to be attached there in its replica about its previous file locality and entire number of replications applied by which all the same existing copies is located and deleted. Most of the organizations are not capable to perform such forensic deletion or destruction of data from the storage and always have susceptibility of data regeneration attacks. Thus this approach gives a concise study of such issues and provides a solution to overcome the existing data devastation issues.

This work defines a problem associated with the automatic destruction with the properties of ZRP and ILM, along with a induced encryption for getting the higher security systems. The system had high system and performance constraints and the exchanges of data are guided with some primitives of object oriented for helping the complete removal with reduced cost. It separates the data and metadata to increase the throughput which reduces the data traffic and increases the processing speed.

II. Background

Data is the most important asset for computing and the other process and techniques used will only facilitate the transitions. All it requires a secure and robust way of handling the data transformation process from different devices, locations, types, copies and networks. Developing a ZRP and ILM based secure approach manages the values changing data and its removal without any footprints. Even metadata and properties will also be removed as the usages are over. The information which is primarily used and open will stays for longer stage and the data with less use will isolated more recurrently. But in existing scenarios there is no such guiding principle obtainable for efficient data demolition. It was named in many ways by designer and researchers like sanitizing, vanishing, disposal, decommissioning, destruction, deletion, removal etc. Getting the complete removal will depends on the changing parameters and when the values is countable then the time is the most important entity of usability. In any system when the fault occurs then the reliable system first generates and transfer the copy of data as replica to some other location or system. In this way the users or any system will generates several replicas of the data. This replica will mislead to the malicious access and will affect the security of the system.

A data physical characteristic shows that the complete removal of the data is very difficult and there exist a probability of its restoration or recreation. It may result in disclosing the sensitive information from its residues which was there in the storage. Thus secure removal from storage systems has become difficult today. The policy links attributes as specified in removal operations to the security classes that must be erased accordingly. The self removal or zero remnance proof based approaches are used to pretend the disclosure of cryptographic keys before the values of data expires. Here the destruction is automatic, without the use of any explicit delete action by any parties involved the data vanish by its own, without need to modify any of the copies of the data, without the use of secure hardware and without relying on new and trusted exterior services, it provides the receiver with the minimum knowledge needed to consume the data.

Thus the automatic destruction or data removal framework can be developed by integrating the multiple key storage mechanism into a single system using object handling. Here the object is created for specified lifecycle usages and will be destroyed automatically after this trigger or counter is over. Here objects are known as active objects for storage having several triggers reflecting their operations after which the complete removal is performed. Some of the prototype based solution is given with [2] and the distributed has table based solution given in [3]. Motivated by the previous system containing the properties of auto removal and zero remnance proof mechanism this work had found that it is possible to generate a system that can permanently remove data after a timeout:

(i) Even if an attacker can retroactively obtain a pristine copy of that data and any relevant persistent cryptographic keys and passphrases from before that timeout, perhaps from stored or archived copies;
(ii) Without the use of any explicit remove action by the user or the parties storing that data;
(iii) Without needing to modify any of the stored or archived copies of that data;
(iv) Without the use of secure hardware; and
(v) Without relying on the introduction of any new external services that would need to be installed (whether trusted or not).

Thus by taking the above goals of work the system founds that the time is the major factors behind the trigger functionality. It also works as vulnerability analysis for nodes and replicating copies of the same clusters. A probing, comparison and verification of pre and post data during this time will yield unwarranted results thus leading to the nullification of the zero data remains proof. Business continuity and availability post geographic calamities etc. An obvious concern of data privacy prediction such a setup inevitably which generally can be overcome by various methodologies like compulsory storing the data in an encrypted form.

III. Literature Survey

During the last few decades the storage technologies had changed come across the way to make the things controlled by the user itself. As of now the storage requirements is exponentially growing which requires self deletion with complete removal so as to make the destruction completely by which the regeneration of data can be avoided. In a way to achieve our goal this work had studied various research articles and included them as surveyed literature.

In the paper [2], an active storage framework is suggested named as OASIS. The paper says that the using object based storage devices is better than any other block oriented media. The active objects manage the rich semantics and object operations with aggregation capabilities. It serves transparent processing along with a complete granularities and permission scalabilities. The results are evaluated on three applications of real world entities and found that OASIS is performing as required.

In the paper [3], data security is handled by controlling the time limits of data disappearance for unauthorized accesses. It is based on peer to peer architecture with distributed enabled communication. Here the data is protected using an encryption standard with distributed hast table based key storages. It uses the VUZE timeout encapsulated with the data objects after the completion of which the complete removal can be performed. Comparison is made with the traditional data destruction applications.

In the paper [4] a novel system FADE is designed for cloud storage which focuses on protecting the removed data using policy based responsive deletion. It is made by the traditional cryptographic standards having high integrity and privacy constructs and has complete data removal properties after the usages are over. It also shows minimum tradeoffs for the performance attributes implemented for Amazon S3 along with other value added features.

In the article [5], a complete scenario is presented for understanding the complete deletion and their associated risk for the solid state disks (SSD). The survey had put some devastating figures and found that 51% companies have to replace their SSDs due to error rates. To solve this problem the organization may require the effective policies for complete and assets disposals. It included the heterogeneous media based policies and how it was written on the disk. Simply deleting all files must have some remaining residuals and which can be recovered and may cause theft. The survey included various tools and service employed for this task. The paper [6] presents an approach named as Vanish whose intension in to remove the data completely. It keeps the track of all the copies of data in P2P system and made their simultaneous removal after the time periods is over. It uses a distributed hash tables as cryptographic standard. The approach ensures that all copies of certain data made unreadable after a defined time boundaries. The paper also presents Vanish, a proof-of-concept prototype based on the Vuze global-scale DHT. Vanish causes sensitive information, such as emails, files, or text messages, to irreversibly self-removal, without any action on the user’s part and without any centralized or trusted system.

Carrying forward the above approach of Vanish and simplified model Safe Vanish is proposed in [7]. This is an improved mechanism by which the data can be able to remove itself after the end of use and increases the privacy parameter. The approach implements a threshold function k for generate the composite key. It sustains the self destructing nature by limiting the attacker's prone zone and sniffing the attacks in actual systems. At the primary work stages and implementation prototypes is proving the efficiency of the suggested approach.

The paper [8] explores the use of encryption standard while developing a new storage hierarchies with time based triggers allotment scheme based on the key. It controls the secure deletion of the data or system keys. The paper uses the concept of policy based secure deletion for erasable memory management and data manipulation operations. The paper also constructed a formal security model along with the threshold based secret sharing but can be extended to public key model. The paper had also shown a prototypic implementation on Linux file system for proving the results of the policy based secure deletion.

The paper [9] gives a framework for automatic, simultaneous self data destruction scheme for object based storage. The approach is an integration of multiple storage techniques used for achieving the self destructions. It uses a new security based storage structure named as Tide using Apache web servers. The approach also shows the comparison of various attack scenarios of Vuze DHT under the given cascading schemes. The approach is an extensible method for supporting the Tide, Vuze and OpenDHT. Results show that the approach is serving up to its defined goal boundaries.

Another self destructing mechanism is given by the paper [10] for cloud applications. It emphasizes on the security parameter by preserving cached copy of memory by using the sanitization methods. Mainly it is guided by the time factors and after its limit is expired the data is automatically removed from all the devices. The approach uses message authentication code along with the OSD features for protecting the malicious uses of metadata. It processes the users request towards the data handling as active object containing the time oriented storage. Another related concept is shown in the paper [11] which uses time constrained data destruction in cloud systems using virtual machines. Here the user’s novel files and message authentication code (MAC) is splitted into several parts and again each part is encrypted with blowfish mechanisms.

Another approach having self destruction is applied using the SeDas phenomenon suggested is paper [12].
Here the data is added with time specific triggers after which the copy of data is made unreadable automatically. Even a destruction key is used here which made the complete removal of data after the defined time limits. It uses an active object where the data is tightly coupled with the time triggers and hence the executable is created for verifying the applicability of the approach during the file access. Compared to the system without self-destructing data mechanism, throughput for uploading and downloading with the projected *SeDas* acceptably decreases by less than 72%, while latency for upload/download operations with self-removal data mechanism increases by less than 60%.

IV. Proposed System

Data Removal with zero remnance proof works towards removing the complete data and its replicas once its lifecycle was over. It covers the complete process from the creation of data to its removal. In today’s world the data is been distributed in several copies and hence to keep track of the replicas and their security is quite a complicated task. Apart from that the data processing requires temporary copy generation which might get removed after the process is completed but sometimes it won’t. Multiple copies and their replica management and removal is performed manually till now which is very tedious job to do. This paper presents a new model for controlling the information lifecycle management using an automated process for secure distribution with complete destruction. Here the destruction mechanism is applies the process of encapsulating the removal parameters with the created objects. Approach is a hybrid combination of multiple triggers integrated with ZRP and hence named as TD-ZRP (Trigger Driven Zero Remnance Proof). Our suggested model serves the factors of security using an RSA based cryptosystem. The data generated by our approach is an executable object which is having an associated trigger attached with it. Triggers will define the initiation conditions after which the complete destruction is called or zero remnance is guaranteed. For authenticating the sender and the receiver the data is having a controlled access which is maintained and recorded using the executable. Once the data is distributed then the removal is assured after the trigger condition is met even without any other interaction from the sender. The approach suggests the two type of trigger i.e. time and read count. The objective with the system is to keep chatting guided be the phenomenon of life cycle. Here the data removal is performed using in memory compilation which overwrites the class flake for that encapsulated data.

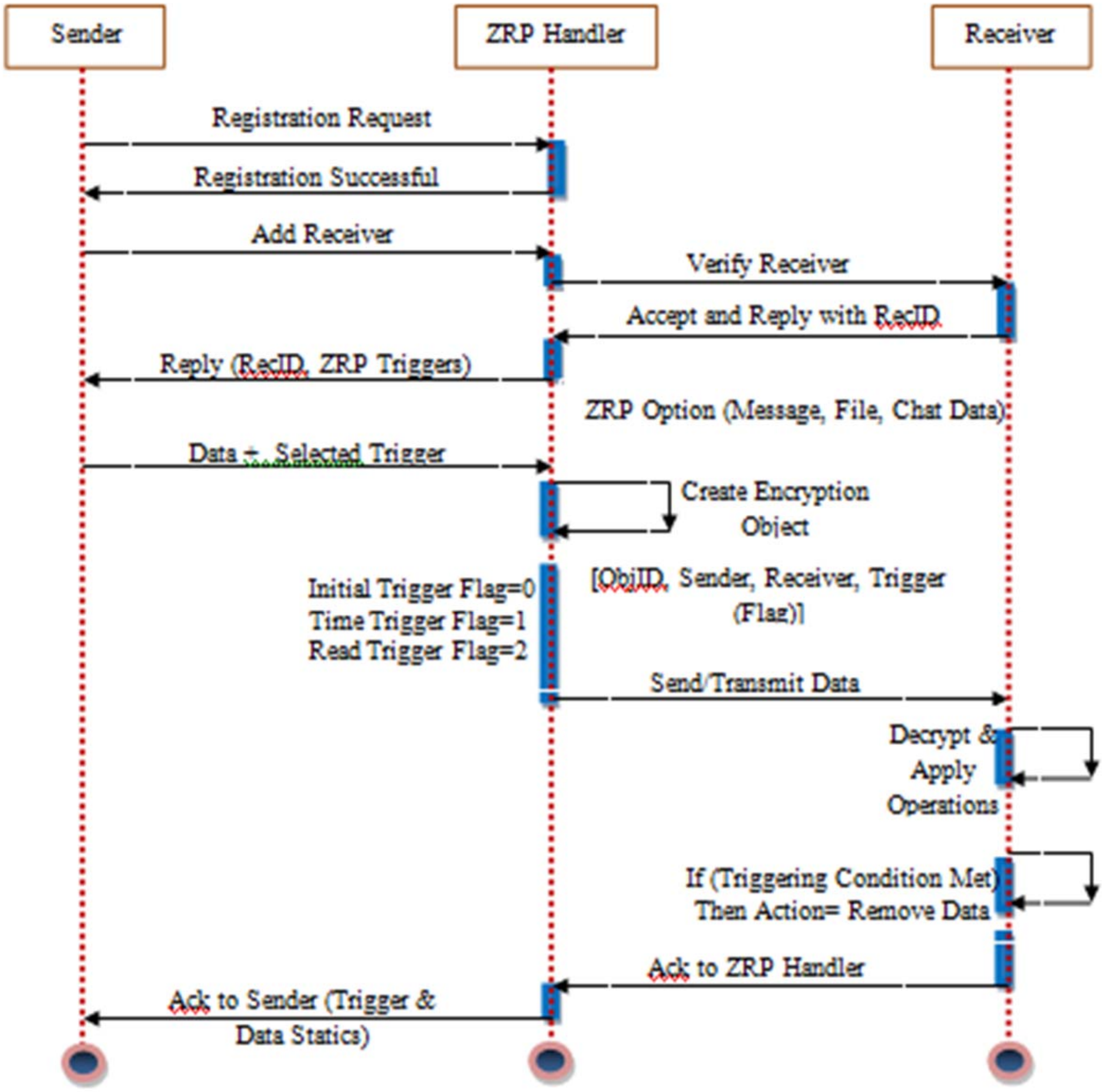
Sender
ZRP Handler
Receiver
Registration Request
Registration Successful
Add Receiver
Verify Receiver
Accept and Reply with RecID
Reply (RecID, ZRP Triggers)
ZRP Option (Message, File, Chat Data):
Data +, Selected Trigger
Create Encryption Object
Initial Trigger Flag=0
Time Trigger Flag=1
Read Trigger Flag=2
[ObjID, Sender, Receiver, Trigger (Flag)]
Send/Transmit Data
Decrypt & Apply Operations
If (Triggering Condition Met) Then Action= Remove Data
Ack to ZRP Handler
Ack to Sender (Trigger & Data Statics)

Figure 1: Proposed Trigger Driven Zero Remnance Proof Technique (TD-ZRP) for Self Data Removal

Description of Process Sequence

The complete process is analyzed and shown in figure 1 which shows the exact sequence of operation against the executable formation for TD-ZRP feasibility. With each and every request for data access an object is created as a mirrored image of data and will be modified and control the changes on the main file. As the request is not to direct primary copy so data theft will be handled. The changes can be made to first copy object and which later on is applied to all replicas. When this active object is created the approach assures its usage duration in terms of count or time limit after which the objects gets destructed automatically and reduces the risk of compromised security. The sequence shown above is having three actors: Sender, Receiver and ZRP Handler. Initially the sender sends the registration request to the ZRP system. The system replies with the successful creation of user accounts. Later on the sender needs to add the authentic receiver to the system. The ZRP handler verifies the receiver with its ID and then reply to the sender back with triggers selection menu.

The sender selects the type of ZRP service along with the trigger (Read or Time). Services offered by the proposed systems are messages, file and chat based lifecycle handling. Once the values for data field and the trigger are selected, then an encryption is performed with RSA cryptosystem. Later on, this encrypted content is attached to its trigger and converted into the executable. Each encryption object is having Object ID, Senders name (ID), Receiver name (ID), Trigger Flag. The triggers are identified using the flag information, for no trigger the flag is set to the value 0, for time trigger it is 1 and for read it is 2. This data or object is transmitted or transferred to the destination machine and then the decryption is performed and the executable starts working by its trigger counts. Once the trigger condition is met, then the removal is performed synchronously to all the copies of data.

Whenever the data usages phase or lifecycle of the data is ended, the self destruction mechanism is called which removes the data completely with all its remnance proofs. Here the networked storage can be used for improved performance over accessing the storage locations. It holds the real data encapsulated in an object with the devastation time. The server calls the removal time or the stored object with devastation time triggers the safe deletion of the data from the locality. The complete safe version of object is stored at the storage locality with fixed destroying time so the copies can’t be produced from this and if it occurs then it destroys the object copy also.

After the final operations of data removal the ZRP transmits the acknowledgement is sent to the sender.

Benefits of Approach:

(i) Supports the redistribution of the data to a non-trusted machine

(ii) Consistent operation and simultaneous read write into a single file.

(iii) Better management of replicated copies their distribution and retrieval records, changes detection and monitoring

(iv) Self destruction and lifecycle based data sustainability for optimize storage

(v) Zero remnance proof based removal to assure complete removal.

(vi) Centrally controlled mechanism with management console

(vii) Object based data handling for instances based execution

(viii) Automatic scheduled destruction after completion of lifetime

(ix) Reduced vulnerability from attack planned to destruct the privacy and security of the systems.

(x) Synchronous operations

Applications

- Sybil/False Identity Attack Detection/Removal
- Automative Removal approach manages the memory utilizations.
- Improved privacy using message and chat removal based on lifecycle factors.
- Online file and change management
- ERP and BI (business intelligence) software’s.
- E-commerce,
- Retail sector
- Transaction System
- Analytical Evaluations etc.

V. Conclusion

Data security has become more and more essential in the Network environment. Cyber forensic for software storage systems are having the wide applicability for various business communities with wide variety of services. . It covers all the aspect which relates with getting the proof of information exchange between various sources. A new approach is introduced for protecting the data privacy from invaders which may attain, through legal or other means. A more effective operational prototype can be developed using the object based file exchanges with attached triggers for implicit operations. Analytical evaluation and the synchronous operations will show the benefits of the approach in near implementations.

Acknowledgment

I would like to sincerely thank Mr. Rakesh Kumar Lodhi, Associate Professor, dept of Computer Science, PCST, for his valuable suggestions and guidance. I would also like to thank to our related faculty members for their comments and inspirations.

References

[1] Technical Report by Privacy Technical Assistance Center, “Best Practices for Data Destruction”, 2014.

[2] Y. Xie, K. Kumar Muniswamy-Reddy, Dan Feng and Others, “Design and Evaluation of Oasis: An Active Storage Framework Based on T10 OSD Standard”, *A presentation on Storage System Research Centre*, 2012.

[3] P. Pilla, “Enhancing Data Security by Making Data Disappear in a P2P Systems”, in Computer Science Department, Oklahoma State University, PP. 1-18, Stillwater.

[4] Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, “FADE: Secure overlay cloud storage with file assured deletion,” *In Proceeding SecureComm*, 2010.

[5] Mr. Sandip N. Vende, Asst. Prof. Nitesh Rastog, "IT-DHSD: IMPLICIT TIME BASED DATA HANDLING AND SELF DESTRUCTION USING ACTIVE STORAGE OBJECT FOR CLOUD", International Journal of Computer Engineering and Technology (IJCET), Volume 5, pp. 01-10 April 2014.

[6] R. Geambasu, Tadayoshi Kohno, Amit A. Levy and Henry M. Levy, "Vanish: Increasing Data Privacy with Self-Destructing Data", University of Washington.

[7] L. Zeng, Z. Shi, S. Xu and D. Feng, "SafeVanish: An Improved Data Self-Destruction for Protecting Data Privacy", *Presentation at CloudCom*, Dec 2013.

[8] C. Cachin, K. Haralambie and H. C. Hsiao, "Policy-based Secure Deletion", at IBM Research, Zurich, Aug 2013.

[9] R. Geambasu, T. Kohno, A. Krishnamurthy, A. Levy and H. Levy, "New Directions for Self-Destructing Data Systems", University of Washington, 2010.

[10] S. Backya and K. Palra, "Declaring Time Parameter to Data in Active Storage Framework", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET),* Volume 2, PP. 3127-3131, December 2013.

[11] M. Nandhini and S. Jenila, "Time Constrained Data Destruction in Cloud", *International Journal of Innovative Research in Computer and Communication Engineering,* Vol.2, PP. 339-343, March 2014.

[12] L. Zeng, S. Chen, Q. Wei and D. Feng, "SeDas: A Self-Destructing Data System Based on Active Storage Framework", *IEEE Transaction on Knowledge and Data Engineering*, Singapore, 2013.