ABSTRACT

Treasure Island Security framework: A Generic Security Framework for public clouds

By Ali Shahbazi

June, 2014

Director of Thesis: Dr. M. H. Nassehzadeh Tabrizi

Major Department: Software Engineering

In this thesis we introduce a generic security framework for public clouds called Treasure Island Security framework that is designed to address the issues related to cloud computing security and specifically key-management in untrusted domains. Nowadays many cloud structure and services are provided but as an inevitable concomitant to these new products, security issues increase rapidly. Availability, integrity of data, lack of trust, confidentiality as well as security issues are also of great importance to cloud computing users; they may be more skeptical of the cloud services when they feel that they might lose the control over their data or the structures that the cloud provided for them.

Because of deferred control of data from customers to cloud providers and unknown number of third parties in between, it is almost impossible to apply traditional security methods. We present our security framework, with distributed key and sequential addressing in a simple abstract mode with a master server and adequate number of chunk servers. We assume a fixed chunk size model for large files and sequentially distribution file system with 4 separated key to decrypt/encrypt file. After reviewing the process, we analyze the Distributed Key and Sequentially Addressing Distributed file system and it's Security Risk Model. The focus of this

thesis is on increasing security in untrusted domain especially in the cloud key management in public cloud. We discuss cryptographic approaches in key-management and suggest a novel cryptographic method for public cloud's key-management system based on forward-secure public key encryption, which supports a non-interactive publicly verifiable secret sharing scheme through a tree access structure. We believe that Treasure Island Security Framework can provide an increased secure environment in untrusted domains, like public cloud, in which users can securely reconstruct their secret-keys (e.g. lost passphrases). Finally, we discuss the advantages and benefits of Cloud Computing Security Framework with Distributed Key and Sequentially Addressing Distributed file system and cryptographic approaches and how it helps to improve the security levels in cloud systems.

Treasure Island Security framework: A Generic Security Framework for public clouds

A Thesis

Presented to the Faculty of the Department of Computer Science

East Carolina University

In Partial Fulfillment of the Requirements for the Degree

Master of Science

by

Ali Shahbazi

June, 2014

Treasure Island Security framework: A Generic Security Framework for public clouds

Images by

Ali Shahbazi

APPROVED BY:

DIRECTOR OF THESIS: _____
Nasseh Tabrizi,PhD

COMMITTEE MEMBER: _____
Mark Hills, PhD

COMMITTEE MEMBER: _____
Junhua Ding, PhD

CHAIR OF THE DEPARTMENT OF COMPUTER SCIENCE:
_____
Karl Abrahamson, PhD

DEAN OF THE GRADUATE SCHOOL:
_____
Paul J. Gemperline, PhD

ACKNOWLEDGMENT

My most sincere thanks go to my advisor and mentor, Dr. Nasseh Tabrizi. I thank him for introducing me to the wonders and frustrations of scientific research. I thank him for his guidance, encouragement and support during the development of this work.

Dr.Tabrizi has been supportive and has given me the freedom to pursue various projects without objection. He has also provided insightful discussions about the research.

I dedicate this thesis to

my family, my beloved wife, Maryam Naghashan, my mother who was my first mentor

for their constant support and unconditional love.

I love you all dearly.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Cloud computing, in its varying incarnations, continues to emerge as an attractive deployment option for enterprises and organizations seeking ways to reduce and better manage the costs associated with application deployment. Cloud providers and users both welcome new cost driven approaches. Commercial cloud services allow organizations to consume computing resources in a manner similar to traditional utilities, like electricity or water; paying for computing resources in a matter commensurate with their use. This Platform as a Service (PaaS) model additionally exteriorize the costs associated with both system administrator and infrastructure administration while providing a potentially more scalable and reliable deployment environment [1]. These significant benefits have created an impression in the minds of many consumers and organizational decision makers that "the cloud" is the answer to any number of software dilemmas.

But while the aforementioned benefits are undoubtedly attractive, these public cloud services are not without significant drawbacks within certain usage scenarios. In circumstances involving highly confidential, sensitive or secret data, security issues inherent to public clouds render their use imprudent, impractical or in some cases impossible depending upon legal and regulatory requirements.

1.1 Cloud Computing Security Issues

Many papers and reports are published every year about cloud security issues and related risks, in this section we recall the concerns that were considered in our previously published paper [33]. Government entities and health care organizations for instance often face legally mandated data security requirements that nearly all cloud services are incapable of satisfying due

to a host of real and perceived security related issues [2-5]. The perception of the security and confidentiality vulnerabilities of public clouds has been reinforced by a number of data breaches reported in the media [6]. While governmental entities, regulatory bodies and medical organizations may benefit from the cloud given the large volumes of data generally involved with their respective activities, the risk of a single data breach often outweighs the potential benefits. Although some cloud providers continue to address these security and regulatory issues, as Microsoft has with the addition of Health Insurance Portability and Accountability Act (HIPAA) compliance features added to its Window's Azure cloud service [7], public clouds still possess too many security unknowns for many organizations.

As a result of these issues hybrid clouds have emerged as a middle ground between the aforementioned benefits of cloud computing and the identified security issues. Private clouds, are developed and administered by an organization's internal IT department for the exclusive use by specific users or user groups within the organization [1, 8]. It is presumed that this greater degree of control guarantees an elimination of the security and regulatory issues posed by public clouds. But these private clouds may suffer from many of the same security issues as public clouds leading to a number of proposals designed to address these issues.

Public clouds offer different advantages like better return on investment, availability, and most importantly, the low cost of use; however there are increased concerns about lack of reliability and security. The important question is 'how to take advantage of the public clouds as an infrastructure to achieve cost saving without sacrificing security and reliability'? In this thesis we introduce Treasure Island Security Framework with two different levels of security models while also investigating different approaches in cloud security like Meta Data Encryption [21], distributed key and sequential addressing [33], mutation and lease that proposed and developed

by GFS group in google [22] [23]. These approaches have their pros and cons and impose additional costs to cloud providers. On the other hand recent research shows that security is the most common concern with cloud costumers [52][53], so a cost driven approach with decent security level could help cloud providers to have more security at a reasonable cost.

Cryptographic techniques are one of most well-known and prevalent techniques in cloud security with key management as an impartible component of any cryptographic techniques. A weak point of the public cloud is in the case of a lost passphrase Secret Key (SK), there is no trusted third-party which can securely recover SK with a well-defined access policy. The recovery mostly runs through a procedure which must be ran by an authorized administrator where even in presence of a good policy and ethical agreements, the administrator has the chance to look at this secret key. Therefore we need a mathematical formulization of security to protect this procedure from adversaries which try to break the security of the encryption scheme. We propose a novel cryptographic approach for public clouds key-management service to avoid this problem in untrusted domains.

1.2 Thesis Contribution

In this thesis, we introduce our security framework titled as the Treasure Island Security Framework (TISF) which builds upon existing thinking to provide a scalable security framework for public clouds. TISF has two security levels,

a. Basic level: TISF takes advantage of distributed key methodology
b. Advance level: TISF uses cryptographic approaches in cloud's key-management.

1.3 Thesis Overview

The remainder of this thesis is as follows:

- Chapter 2 will outline the cloud computing and security, definitions and approaches.
- Chapter 3 will present the related works and discuss about other papers and researches in this area
- Chapter 4 will all about our proposed security framework, Treasure Island Security Framework, and it's security level
- Chapter 5 will explain the security proof for our model.
- Chapter 6 will summarize the main points of the thesis and will show possible new directions for future works.

CHAPTER 2: BAKGROUND

Cloud computing might be a new name of technology which hints at a future in which chromebooks and similar cloud-based computers will supersede local computers, but believe it or not it is not a new idea; and appears with different names like grid computing, etc. John McCarthy, an American computer scientist who coined the term “Artificial Intelligence” predicted in 1961 that “computation may someday be organized as a public utility”. In the early 90’s his idea was followed by Joseph Carl Robnett Licklider’s who brought the “intergalactic computer network” idea in which “for everyone on the globe to be interconnected and be able access programs and data at any site, from anywhere”.

The term of “Cloud Computing” is used by Ramnath Chellappa [29], [30] in his speech, “Computing has evolved from a main-frame-based structure to a network-based architecture. While many terms have appeared to describe these new forms, the advent of electronic commerce has led to the emergence of 'cloud computing.' This work aims at analyzing the role of agents and intermediaries enabling this framework.”

One of the best official definitions of cloud computing provided by The National Institute of Standards and Technology (NIST) [31]”Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.”

These days Cloud computing will be offered in different categories by different providers and companies with three main service models; but most of their services and models are similar in these characteristics:

- On-demand self-service which allows users to use computer capabilities like network storage or server time that are prepared unilaterally by their Cloud service provider
- Rapid elasticity: this is one of major characteristics which sets cloud computing apart from other concepts like data centers and networks of networks in that Capabilities can be elastically provisioned and released to scale outward and inward commensurate rapidly mostly automatically [31].
- Resource pooling: users of cloud computing can use the ability of a cloud service to serve multiple customers by using a multi-tenant model. These different models and services could be on different physical and virtual resources which dynamically assign and reassign according to demand [31].
- Measured service: all systems, resources and services will be controlled and optimized automatically by the cloud system, and these resource usages can be easily monitored and reported by the cloud provider.

Cloud Computing premises can be broken up into these three main Categories.

- Software as a Service (SaaS)
- Infrastructure as a Service (IaaS)
- Platform as a service (PaaS)

Software as a Service (SaaS)

In this model the application delivered by the cloud system is ‘as a service’ rather than ‘as on-promise’ software. Various client devices have access to applications through a program interface or even a thin client interface like web browsers (or web-based email). Software as a Service is becoming a progressively customary delivery model as underlying technologies

support web services and SOA, new scripting, developmental approaches. This popularity was accompanied by rapid growth of broadband services that is widely available to support user access from more locations around the world.

The key fact of this popularity is described by NIST [31], "The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings."

Easier administration, Automatic updates and patch management, Compatibility, Easier collaboration, and global accessibility are some of benefits of SaaS model.

Platform as a Service (PaaS)

In the Platform as a Service delivery model, cloud providers will prepare all hardware, operating systems, network capacity and storage for the user over the internet. PaaS will allow the customer the ability to rent virtualized servers and relevant services for running existing applications or developing and testing new applications through the internet.

In this model the customer will not be required to manage or control the underlying cloud infrastructures like storage, servers, networks or even individual application capabilities (with the possible exception of limited user-specific application configuration settings). [31]

Infrastructure as a Service (IaaS)

Infrastructure as a Service is a provision model in which an organization outsources the equipment and hardware used to support operations (sometimes is addressed as hardware as a service). The cloud provider is responsible for ownership, maintenance and running the services. The client typically pays on a per-use basis for components of IaaS that mainly include utility

computing service and billing model, automation of administrative tasks, desktop virtualization, dynamic scaling and policy-based services.

By the definition that NIST provide [31] “The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components”.

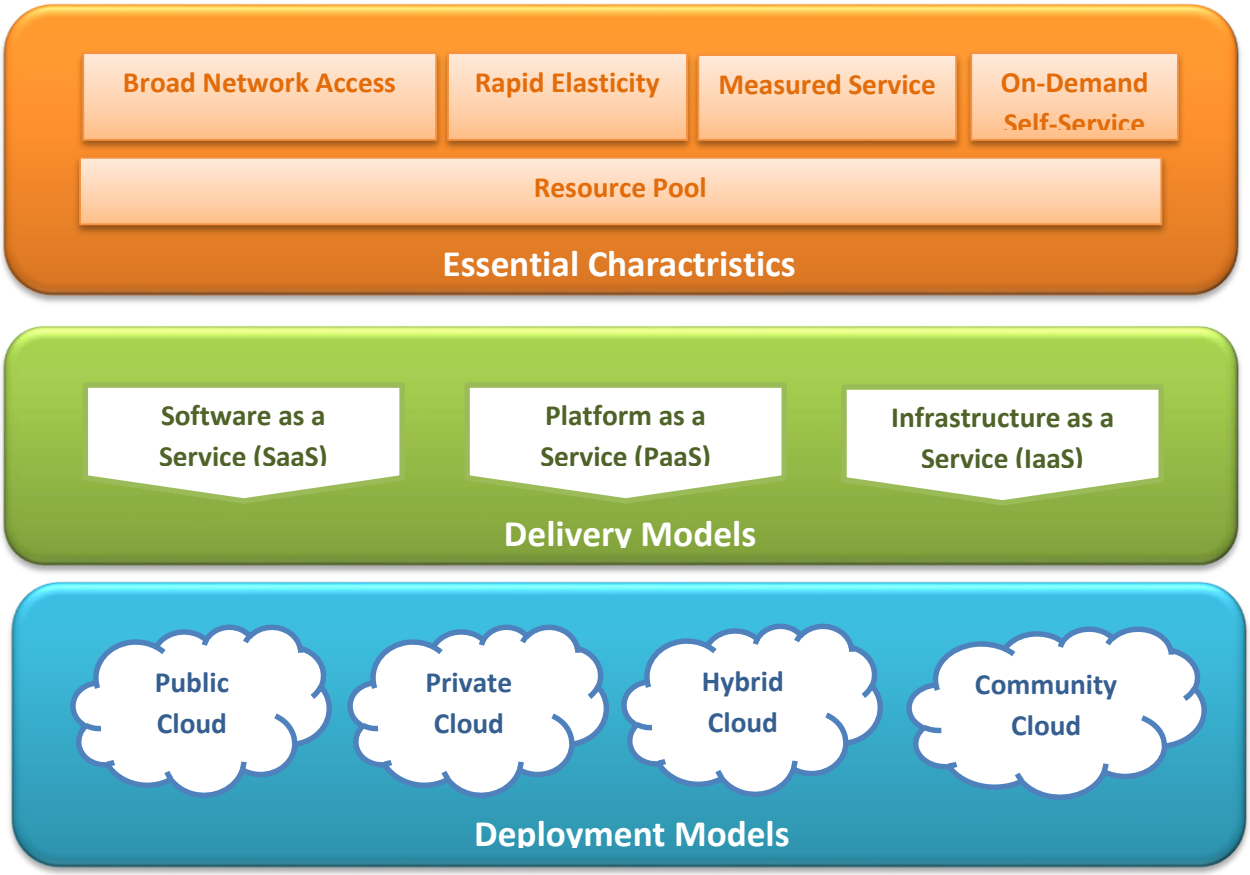

Figure 1. *Cloud computing different layers and approaches based on NIST definitions.*

CHAPTER 3: RELATED WORKS

There is a significant body of work which documents the challenges and proposed solutions to the issue of cloud security; in both public and private [9-11]. We recall some of these related works that were previously addressed in our research [33]. Many of these works take substantially different approaches to the issue of cloud security given the broad topic that is cloud computing. The approach taken within these works revolves primarily around an overlapping use of encryption, distributed key methodology, sequential chunk addressing and dynamic metadata reconstruction to improve system security. We consider related works to both distributed key and cryptographic key management in this chapter.

3.1 Distributed Key with Sequential Addressing

Distributed key methodology is not a new concept, having significant support with the literature albeit in significantly different conceptualizations and implementations [12-14]. A form of distributed key methodology serves as the backbone of the security effort proposed within this work whereby the key necessary to decrypt individual file chunks and reconstruct a stored file is distributed within our proposed system. This methodology stands in contrast to the some of the more common methods of cloud authentication such as those based primarily on password protection and Private Key Infrastructure (PKI). While these techniques are relatively easy to implement, they have a number of deficiencies which have been documented in the scientific literature and the media. Password protection for instance is dependent upon the user's ability to maintain confidential information against social engineering attacks wherein information enabling

the reconstruction of one or more passwords may be divulged inadvertently by a user [15]. Password methodology is additionally troublesome given the average user's penchant for password reuse [16]. While the reuse of an existing password minimizes the cognitive load that memorizing a number of different passwords for different systems creates, reuse effectively means that a password compromised for one system allows malicious users to access a host of other user accounts. This is especially troublesome for email accounts which often serve as key link in the user verification process for many systems. Ticket based authentication using the Kerberos protocol is an improvement over pure password based protection, however, this methodology still possesses security risks [17]. A breach of a system's authentication server will result in the exposure of all user accounts due to the centralization of authentication management. [18]. Public Key Infrastructure (PKI) addresses some of these issues through the use of digital certificates for entity identity verification [19]. This approach however also has a number of flaws [20]. The distributed key approach utilized within our proposed system addresses many of these issues by using a decentralized form of authentication that eliminates the single point of failure found in password protection scheme given the use of a segmented, dispersed key. This methodology is further bolstered by our use of a form of dynamic metadata reconstruction, which protects information about the stored data and chunk encryption [20].

3.2 Cryptographic Key-Management

Cryptographic approach has a long history of use in the clouds security, each part of our proposed model, if considered alone, have a lots of similar related works. Our approach in relying on Public Key Encryption and Forward-Secure Public Key Encryption and PVSS

schemes to access trees, and heavily based on approaches that had been introduced by d'Souza et.al [34].

We introduce Forward Secure Public Key Encryption which support a non-interactive Publicly Verifiable Secret Sharing Scheme, to take advantage of session based PKE scheme which is more secure than a simple PKE scheme. Using FS-PKE which is constructed based on Binary Tree Encryption method, will help us to compare the FS-PKE + PVSS model with other types of forward security schemes like Hierarchal Identity-Based Encryption methods.

The idea of forward security for public key encryption was introduced by Ross Anderson in 1997 during an ACM conference on Computer and Communication Security [47]. FS-PKE is based on session keys which are secured for different sessions independently, and in case of compromising one key other sessions will not be affected.

There are different approaches, like weak forward security, that take advantage of security mediator [48] [49] which holds a share of secret key and both user and security mediator must sign on a message or decrypt the received cipher-text. Strong forward security [50] is another approach which improves the security issues that appear in weak forward security by updating public key in each time period by user before sending message to CA to get a certification. Both weak and strong forward security schemes are based on time sessions.

In Publicly Verifiable Secret Sharing, D'Souza [34] shows that only a standard PVSS scheme could not be supported by a PKE scheme, they worked on Stadler [51] and standard extension of blakley-shamir secret sharing scheme [34] to take advantage of using the Access tree in their work. Our approach heavily relies on their work, but we prefer to use another non-interactive PVSS scheme, which satisfy BDH conditions.

In Next chapter we will introduce Treasure Island Security Framework, which mostly focused on the role of keys in cloud security.

CHAPTER 4: TREASURE ISLAND SECURITY FRAMEWORK

While the Google File System (GFS) [22-23] and the Hadoop Distributed File System (HDFS) [24-26], a GFS derivative, are commonly utilized with public clouds, we have proposed an alternative cloud architecture upon which the Treasure Island Security Framework is based. The Distributed Key and Sequentially Addressing Distributed file system (DKASA), which builds upon aspects of both GFS and HDFS, improves the security of data storage and file distribution in a private cloud primarily through the introduction of dynamic metadata reconstruction, sequential addressing and distributed key methodology. Later in this chapter we will introduce our cryptographic approach, based on FS-PKE which support a non-interactive publicly verifiable secret sharing scheme through an access tree. This security level which takes advantage of individual sessions, can be safe and secure approach in public clouds key-managements or generally for any key-management service who act in an untrusted domain. These approaches can help user to avoid the common security related harms in public clouds like data loss, inaccessibility, replication or data modification and access which causes by threats like hacking, government powers or injection. Table 3.1 summarizes these harms and threats:

Table 3.1. Cloud computing threats and harms

Threat / Harm	Hacking	Government Powers	Injection	DDoS attack
Data Loss	√	√	√	X
Inaccessibility	√	√	√	√
Data Modification	√	√	√	X
Data Replication	√	√	√	X

4.1 Distributed Key and Sequentially Addressing File System

As we introduced in our previously published paper [33] the Distributed Key and Sequentially Addressing Distributed file system (DKASA), as illustrated in Fig. 1, has a number of characteristics borrowed from the GFS including a single master configuration, use of fixed chunk sizes and chunk replication. We provide assumptions with respect to the configuration of DKASA within our proposed framework to contextual the security risk model discussed in Chapter 5. We first recall the definition of some elements from our recent paper [33].

4.1.1 Single Master Server

Both Single Master (SM) and Dual Main Server (DMS) configurations are possible within the DKASA file system although our method is based on the use of the former; mirroring the GFS. The DMS configuration, which involves the use of a management server and a file retrieval server, is potentially less robust than the SM configuration given its relatively poor performance under stress. It is likely, with moderate to high levels of network traffic and significant numbers of large files, that the retrieval server in the DMS configuration becomes a bottleneck for the entire system.

4.1.2 Fixed Chunk Size and Multiple Replicas

We anticipate the use of a fixed chunk size, which enables the use of the GFS mutation and lease method to reduce network traffic. Unlike the GFS which uses a fixed size of 64MB we have consciously chosen to leave the size of the chunk ambiguous as both large and small chunk sizes have advantages and disadvantages. A large chunk size for instance will reduce the number of chunk servers needed for each client while also reducing the client's interaction with the master for reading metadata and namespaces. This large chunk size however is also incompatible with smaller files. A smaller chunk size is compatible with smaller files however it may result in

greater data fragmentation. In an actual implementation of our framework a system architect would determine the appropriate chunk size for the specific usage scenario.

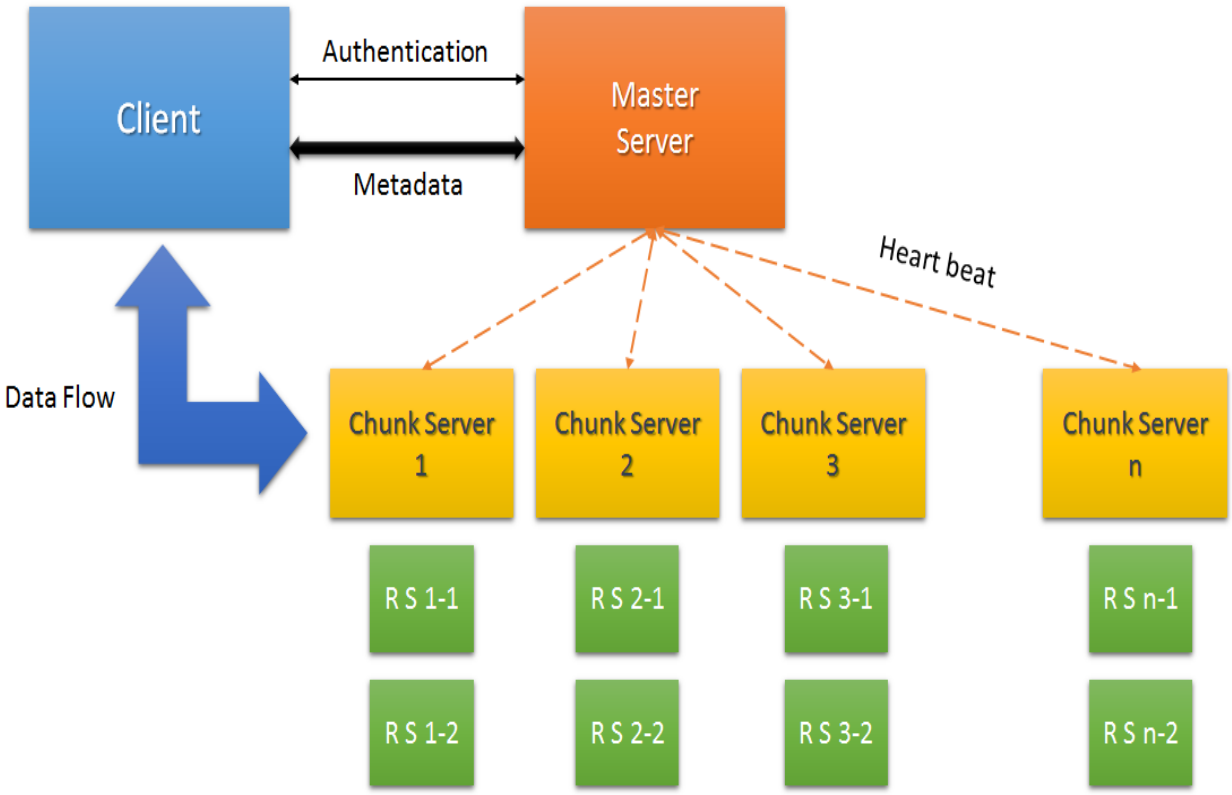

Figure 2. *Treasure Island Security Framework Abstract Model*

Each chunk within the system will have a number of replicas (k) to ensure data availability; each replica chunk exists in isolation from the original. In the event that the original chunk is unavailable the replica system will retrieve a replica and use that data to reconstruct the original file.

4.1.3 Encryption

Each chunk's data is encrypted in the client machine and sent via secure communication using RSA [26] and Advanced Encryption Standard (AES) encryption [28]. The key differentiation

between traditional approaches and our approach that proposed in this work is the use of a distributed key approach as opposed to the use of usual Public Key Infrastructure (PKI).

4.1.4 Distributed Key and Sequential Addressing

The use of a distributed key methodology has been driven mainly by two important factors, a desire to increase security while introducing a level of tractability whereby different security levels with different costs and different requirements exist within the system. In terms of the latter, this type of security granularity facilitates varying degrees of file and user level security as opposed to a methodology within which the level of security within the system as whole is the only manipulable value.

Distributed key methodology as proposed within the DKASA system involves the distribution of a cryptographic key into four isolated parts. The first two parts of the key are stored in the master server and the client, the third part is stored in each chunk (n) and the final part of the key is stored in the previous chunk (n -1). For the first chuck the last key will come from Chunk n (last chunk). A file to be stored in the public/hybrid cloud will be divided into a series of sequentially addressed chunks with distinct, appended headers and footers. The header of each encrypted chunk contains the following information:

- 128 bit local deciphering key
- 128 bit remote deciphering
- key The address of the next chunk
- 128 bit status code used to identify the chunk as either an original or replica chunk.
- 1024 bits of audit data

This header data is used by the Cloud Management Server (CMS) and the user's client during the file retrieval process to locate file chunks, decipher them and rebuild the original file using the distributed key and sequential addressing approach. The full key necessary to decrypt each encrypted chunk is produced as a result of the concatenation of the parts of the key stored on the master server, the client, the current chunk server and the previous chunk server are illustrated in Fig. 2.

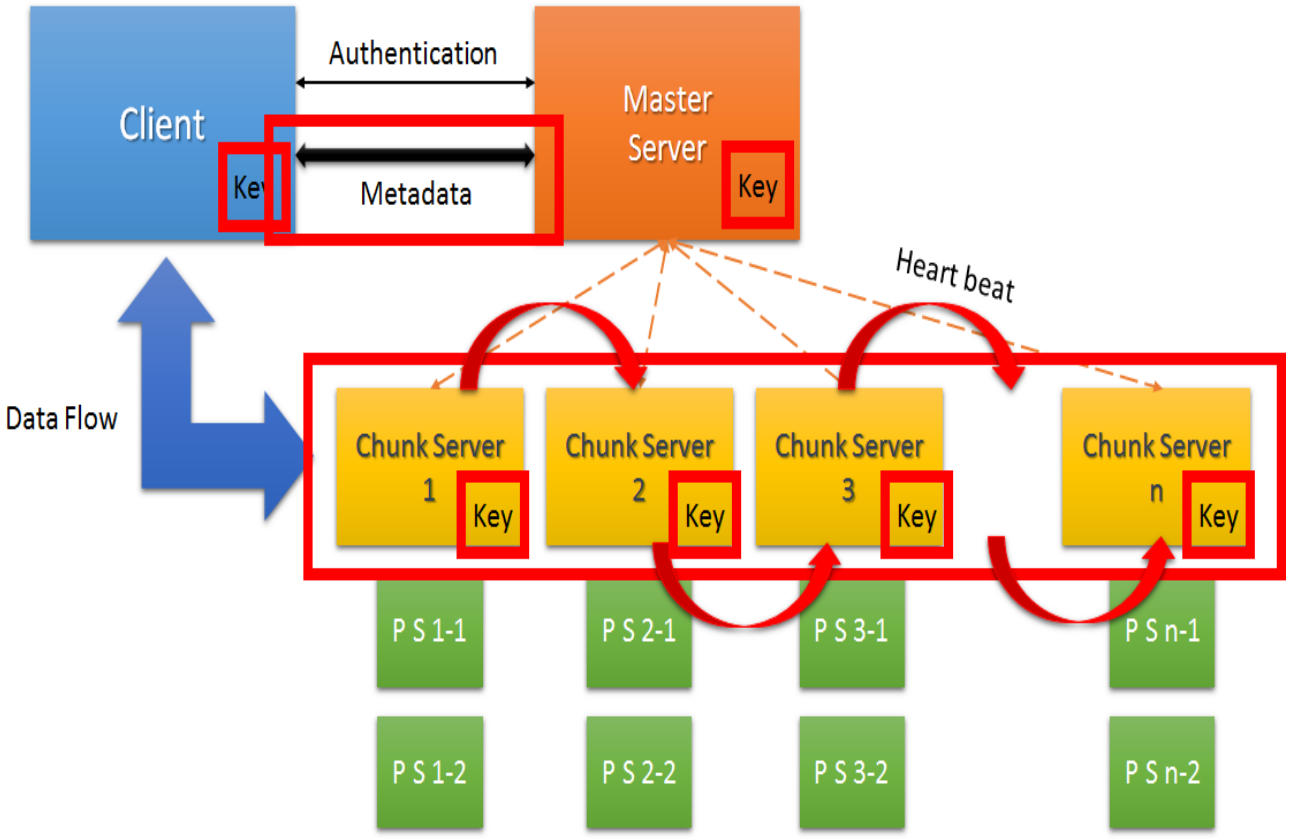

Figure 3. *TISF Sequential Addressing and Distributed Key Methodology*

Upon successful completion of this process an interim copy of the file is available to the user on the client machine. Upon the completion of file manipulation (read, update, delete, etc) by the user,

the mutation and lease method as employed within the GFS the chunks are stored on new servers. The complete file access algorithm is shown in Fig. 3

```
procedure FileRetrieval()
1:    Client machine sends authentication request to master server
2:    Master server checks and approves client
3:    Master server sends SID and service lists to client
4:    Client asks master server for the address of the first chunk
5:    Master server sends first chunk server address and first chunk (n-1)
      code part to client
6:    Client send chunk server request, 128 bits of deciphering key, code
      part (n-1) and first chunk address
7:    Based on internal algorithm client partition is reinterpreted to new
      deciphering code
8:    Client reads and deciphers first chunk from the file server based on
      the reconstructed key
9:     LOOP: Client refers to the next server based on the read data from the
      current server
10:           Client reads and deciphers chunk from the file
              server based on the reconstructed key
11:           IF: File is complete
                  END LOOP
12:           END IF
13:   END LOOP:
end FileRetrieval
```

Figure 4. *The complete file access algorithm*

4.2 Cryptographic Approaches for Cloud Key-Management Service

In this chapter we propose a novel cryptographic approach for public clouds key-management or generally key-management service in any untrusted domain. We will use a forward security public key encryption scheme which must support a non-interactive publicly verifiable secret sharing scheme through an access tree structure.

Our focus in Cloud Key-Management service is on key recovery part. Suppose that user A have pair of (PK, SK) as public key and secret key. If A loses the SK, the system must recover this key through an access structure and verify that this key will works as a secret key. In untrusted domain, a set of procedures will recover and verify the secret key and crucial point is that adversaries can attack and find the secret key during recovery and verification, we must find a way to store any secret keys in encrypted mode (Using PKE) and we need a cryptographic tools to allow us to securely recover and verify this encrypted secret key. For the first part we introduce a FS-PKE scheme and will use a non-interactive PVSS scheme to support recovery and verification.

4.2.1 Preliminaries

Our work is mostly focused on using Forward Secure Public Key Encryption scheme which is supported by a non-interactive Publicly Verifiable Secret Sharing scheme in Cloud-based key management. This method help us to recover a Secret Key (SK) securely without allowing any adversary to find any information about SK. We assume familiarity with definition of PKE, PVSS and EL-Gamal systems.

As we mentioned, this chapter mainly focused on key recovery and verification. We need to store SK in some encrypted form and allow the cloud system to verify that legitimate parties can recover

the secret key without compromising any information about it. For recovery and verification key though a set of legitimate parties we need a tool like publicly verifiable secret sharing scheme [51] and to store SK in an encryption form, we can use Forward Secure Public Key Encryption, which is take advantages from acting as an individual session based cryptosystem.

We will review some of these definitions in the next section:

Access Structure: As reported in [34] [35] [36] a collection $\mathcal{A} \subseteq 2^{\{P_1,,,P_n\}}$ when $\{P_1,,,P_n\}$ is a set of parties, called monotone if $\forall B,C, if \ B \in \mathcal{A}, B \subseteq C \, then \, C \in \mathcal{A}$, An access structure is a collection $\mathcal{A}$, $\mathcal{A} \subseteq 2^{\{P_1,,,P_n\}}$ and $\mathcal{A} \neq \varnothing$,any sets in $\mathcal{A}$ called Authorized and any set not in $\mathcal{A}$ will be called unauthorized.

Binary Tree Public Key encryption (BTE) scheme: Is a 4 tuple of PPT algorithms (Gen, Der, Enc, Dec} [32] [37] such that:

- The Key generation algorithm Gen() must generate public key PK from a security parameter 1^k and initial root Secret Key SK_e
- The Key Derivation algorithm Der() takes 2 parameter, name of a node $w \in \{0,1\}^*$ and nodes associated secret key SK_w and returns two secret keys SK_{w0} and SK_{w1} for the children of node w.
- The Encryption Algorithm; Enc (), will encrypt message m with public key PK and node w and return the cipher text c.
- The Decryption Algorithm; Dec (), will decrypt cipher text c with Public Key PK and node name w.

In the article published by Canetti et.al [32] the correctness requirement, for any key set {PK, SK_e} which have been generated by Gen () Algorithm, for any node $w \in \{0,1\}^*$, and SK_w which are generated for node w, m can calculated as we shown on Eq.(1)

$$m = Dec(\mathrm{PK}, \mathrm{w}, \mathrm{SK}_w, Enc(\mathrm{PK}, \mathrm{w}, \mathrm{m})) \quad (1)$$

Where m is our message, SK_w is the secret key in node w and PK is the public key.

Our contribution includes

- a Forward Secure Public Key Encryption scheme,
- In regards to FS-PKE we use a Public Verifiable Secret Sharing scheme which could be supported by our FS-PKE scheme, over an access structure A.

We assume that there is a Valid-Check algorithm such that shown in Eq. (2)

$$Valid - Check(\mathrm{PK}, \mathrm{SK}) = 1 \text{ iff } \exists \mathrm{k} \in \mathbb{N},\ (PK, SK)\ in\ the\ range\ of\ K(1^k) \quad (2)$$

Where PK is the public key and SK is the secret key

Before introducing our model, it is necessary to look at different parts of this model:

Forward Secure Encryption: in 1989 Gunther proposed a session key exchange protocol and in his paper titled "An identity-based key-exchange protocol" [38], he proposed a term called "perfect forward secrecy", in which for the impersonated user, the hardness of finding the key is equal to breaking Diffie-Hellman scheme for every 3rd-party when the protocol is running. Generally speaking, forward secure encryption is about using session keys. If long term key is compromised the protocol will provide a forward secrecy, and it is independent of past session keys which had been established before the compromise of long term key. This independency between session keys is the biggest advantage of the Forward Secure Encryption.

From encryption point of view, forward security means any break-in to the system has no effect on information which is encrypted prior to break-in. for example consider this sequence actions:

- First, Alice and Bob must generate a shared session key K together,
- Alice encrypt message m with K
- In last step both of them will delete shared session K promptly.

We can define this concept for public or private key encryption respectively, in this situation obviously the confidentiality of information that has been encrypted using some secret info in the past is not affected by loss of secret at present.

In terms of Forward Secure Encryption we mostly follow the notation and approaches that are introduced by Canetti et. al[32], we use Binary Tree Encryption (BTE) definition and introduce four functions of encryption, derivation/update, Decryption and generation. Then we will use the output PK and SK in PVSS model.

FS-PKE supports Publicly VSS: a forward secure public-key encryption scheme supporting publicly verifiable secret sharing for an access structure A consist of 8 function

$$\{Gen, Upd, Enc, Dec, Setup, Share, Verify, \mathrm{Re}const\} \quad (3)$$

Such that the 4-tuple {Gen, Upd, Enc, Dec} is a forward-secure public-key encryption scheme and

Setup (n, 1k) as a randomized algorithm will compute public value P_{P_i} for every $i_{\varepsilon}[n]$ and corresponding Secret Key SKi, the setup function's output is a vector like $\{(P_{P_1}, SK_1), ..., (P_{P_n}, SK_n)\}$

Share (PK, SK, A) is a randomized algorithm for generating encrypted shares, takes PK , Secret Key SK and Access Structure A as input and will produce string π as output.

Verify (PK, π, A) is verification algorithm which takes public-key PK, String π and Access structure A as input and just {1, 0} as output,

Reconst(PK, π, A, SKs) is for recovering Secure Key SK, the function takes Public key and string

$$\{\forall k \in \mathbb{N}, \text{Valid Access Structure A}, P[\text{Verify}(PK, \pi, A) = 1 : (PK, SK) \leftarrow K(1^k) AND \pi \leftarrow Share(PK, SK, A)] = 1$$

π and Access structure A and SKs and the output is new Secure Key SK'.

In security models and VSS scheme we introduce the model that was originally used by D'Souza et.at [34] in their approaches in their PKE-PVSS model and then we introduce new non-interactive PVSS scheme for our FS-PKE & PVSS model which was proposed by Tian et al [39] [40].

4.2.2 Access Tree

In security systems, wherever we have parties which need to work together to obtain a resource, we need an access structure. A qualified set, in fact, is a group of these parties that are granted access. We have different types of access structures, for example monotone and non-monotone access structure (in monotone access structure if subset M is in access structure, all sets A which $M \subseteq A$ is monotone and in access structure), multi-threshold access structure that was introduced by Farras et.al [41] which is suitable for multi-secret sharing schemes and access structures based on graphs that were proposed by Beimel et.al [42]. Here we use a tree threshold access structure that was recently used in identity based encryption [43] [44], attribute based encryption [45] [46] and also verifiable secret sharing. We recall the basic framework of Access Trees that have been used in secret sharing.

Let Access Tree τ be a tree that represented an access structure, each non-leaf node of the tree represents a threshold gate, described by its children and a threshold value, if numx is the number of children of a node x and kx is it's threshold value, then $0 \leq k_x \leq num_x$ .When k_x =1 the threshold gate is an OR gate and when $k_x = num_x$, it is and *AND* gate. Each leaf node of tree is described by a party P_i and a threshold value k_x.

To facilitate working with the access tree, we defined a few functions. We denote the parent of node x by *Parent(x)*. The access tree τ defines an ordering between children of every node. The

children of a node x are numbered from 1 to num_x. The function *ind(x)* returns such a number that is associated with the node *x*.

Satisfying an Access Tree. Let τ be an access tree with root *r*. Denote τ_x by the sub-tree of τ that rooted at the node x. Hence τ is the same as τ_r. If a set $\gamma \subseteq [n]$ of indices satisfies the access tree τ_x, we denote it as $\tau_x(\gamma) = 1$. We compute $\tau_x(\gamma)$ recursively as follows. If *x* is a non-leaf node, evaluate $\tau_{x'}(\gamma) = 1$ for all children x' of node x. $\tau_x(\gamma)$ return 1 if and only if at least k_x children return 1. If x is a leaf node, then $\tau_x(\gamma)$ return 1 if and only if $ind(x) \in \gamma$.

4.2.3 An Efficient scheme FS-PKE support a non-interactive PVSS Scheme

Now we can present our complete system, Forward Secure Public Key Encryption (over a BTE) which defined session based, without loss of generality, we can reconstruct any SK_i (for any *i* in time indices) because for any *i* we have unique SK_i, and Valid () function could test this situation for us. Also we can observe that our FS-PKE scheme is an El-Gamal system [32].

Our system contain a FS-PKE scheme which will support a non-interactive publicly verifiable secret sharing scheme through an access tree. This system is based on bilinear pairing and is proven secure in standard model under BDH assumption. As noted before, we will present an FS-PKE scheme with four functions and briefly talk about non-interactive PVSS schemes which is compatible with our assumption.

The encryption scheme has 4 functions, key generation or Gen () must generate public and secret key, assume 2 groups like G1 and G2 prime order q and also e as bilinear map,

$PK = (G_1, G_2, \hat{e}, P, Q, t, H)$, S_e=αH(ε) where P is a random generator $P \in G_1$ and α is a random variable such $\alpha \in \mathbb{Z}_q$ and set Q= α P

Upd () will update secret keys and because our focus is on reconstructing secret key in untrusted domains we can assume that updating keys do not change anything in reconstructing procedure because we can reconstruct any secret keys in our BTE tree based on our access structure and reconstruction method.

Enc (PK,w,m) [32] will encrypt based on w=w1,….,wl and random $\gamma \in \mathbb{Z}_q$

$\bar{C} = (\gamma P, \gamma H(\omega|1), ..., \gamma H(\omega), \mathrm{m.d})$ and $\mathrm{d} = \hat{\mathrm{e}}(\mathrm{Q}, \mathrm{H}(\varepsilon))^{\gamma}$.

We first recall the decryption function from Canetti et.al [32]. Then, based on this function, we can easily shape our reconstruction method in PVSS scheme. Obviously reconstruction of an encrypted secret key exposes no information about the secret key.

Dec (PK, ω,SKω,C) will decrypt cipher text C and takes ω=$\omega_1 \ldots \omega_t$ and SK_ω =($R_{\omega|1}$,.., R_ω, S_ω) and cipher text C is equal to ($U_0, \ldots, U_l, \upsilon$) and decryption will compute by m= υ/d where d is calculate as shown in Eq.(4)

$$d = \frac{\hat{e}(U_0, S\omega)}{\prod_{i=1}^{\ell} \hat{e}(R_{\omega|1}, U_i)} \qquad (4)$$

Now we can move to the PVSS model and the 4 functions (setup, Share, Verify and Reconst) in PVSS scheme, as you can see in decrypt and encrypt function in our fs-PKE scheme our functions are based on $\hat{\mathrm{e}}(\mathrm{Q}, \mathrm{H}(\varepsilon))^{\gamma}$, $\hat{e}(U_0, S\omega)$ and product of e(,) so we can use this feature and Lagrange interpolation to build (or change) our reconstruction method. The difference between the reconstruction method in our approach and reconstruction in a regular non-interactive PVSS

scheme, like Tian et. al [39] [40], is in our approach we must decrypt shares for relevant leaf nodes in our Access Tree and then apply PPT interpolation in exponent recursive mode to reconstruct the SK so we have an extra step based on our access tree to obtain our reconstruction, In the reconstruction method we must define a recursive function DecryptNode(π,SKs,x) that takes secret key, string π and node x, and the output must be null or an element in G1, so we can easily reconstruct our secret key.

CHAPTER 5: SECURITY ANALYSIS AND VALIDATION

5.1 Treasure Island Security Framework's Security Risk Model

Here we recall the security risk models that was considered before in Treasure Island Framework [33]. This approach adds security beyond that found in security schemes using public key infrastructure and single password methodologies in that the likelihood of system compromise from a single attack is largely eliminated. We use the previously outlined assumptions and the presumption of a single file broken down into many chunks (N), each possessing many replicas (K) within a distributed key architecture to evaluate this claim. We define the distributed key as Eq. (5)

$$D_k = D_i + D_{i-1} + D_{cm} + D_{ms} \qquad (5)$$

Where D_k is Final K, D_i is Key of Chunk i, D_{cm} is Client's Machine Key and D_{ms} represent Master Server's Key

All four isolated parts of the key are necessary to construct the full cryptographic key required to decipher each chunk. To evaluate the likelihood of comprise, it is thus necessary to calculate the availability of each of the four key components with respect to their individual locations: the master server, the client, the original chunk and the previous chunk. Within the working system, the master server is constantly operational, therefore the availability of this component to an attacker is equal to 100% or $Pa[D_{ms}] = 1$

For the client machine the window for an attack is based on the total time of connection. For the purpose of this analysis we assume a client connection duration that is represented as TCM. For the final two key components, it is necessary for an attacker to successfully attack two different chunks, the current chunk and the previous chunk, to assemble all of the components necessary

to reconstruct the full cryptographic key. The probability of a successful attack in this scenario can calculate by Eq. (6)

$$C_n = \frac{(N_s - b)^2 - 2}{\frac{N_s!2!}{(N_s - 2)!}} \qquad (6)$$

Where N_s is total number of Chunk servers and the denominator is the number of all choices and the numerator is the likelihood that an attacker successfully selects the server containing the first chunk. Thus the chance of gaining access to all the necessary items for deciphering a chunk will be (Eq. (7))

$$P_{FA} = \frac{(N_s - b)^2 - 2}{\frac{N_s!2!}{(N_s - 2)!}} \times \frac{T_{CM}}{86400} \qquad (7)$$

Where N_s is total number of Chunk servers

The availability of a file for a user, in the event of a server failure or malware attack, depends upon the number of replica chunks that exist for each original. When there are replica chunks, a $K < n$ high risk situation exists for the integrity of the user data. In that file, chunks exist without a backup. We assume K = n and all chunks have a backup, therefore a full copy of the entire file exists. The total number of full file backups may be determined by $\left(\frac{K}{n+K}\right)$

The chance of successful file retrieval after any malware attack server failure is shown in Eq. (8):

$$A_i = K\frac{P_p}{n+K} \qquad (8)$$

Where P_p equals availability of a server in the system, K the quantity of replica control chunks, and n the number of chunks per file.

5.2 Cryptographic approach security model

For PVSS scheme we need to prove correctness and soundness properties such that for Honest Dealer and shareholders, every qualified subset of shareholders (which identified with access structure A) can reconstruct the secret during the reconstruction algorithm (correctness). We can claim consistency of secret commitment values with share deriving values only if dealer passed the verification algorithm successfully (soundness).

For privacy proof we must show that in presence of honest dealer, adversary cannot learn any information about secret, we defined security through a game played by challenger and PPT adversary, both of them given λ as security parameter.

Just notice that the adversary is static which have the corrupted shareholder's secret key, we could explain this level in 3 steps: Setup, Challenge and Guess.

Setup: The challenger runs the Setup (λ) to obtain the set of public parameters as well as all shareholders Public and Secret keys. In this step, adversary is given t-1 corrupted shareholder's secret keys and moreover all public keys

Challenge: Based on our model, here the challenger will pick two distinct message M and M' with same length and also a random bit b and compute the cipher-text Mb with Share () function and distribute all resulting info to A as well as {M, M'}

Guess: In this step the adversary must output a guess bit b'

The advantage of adversary in this game is defined by Eq. (9):

$$Adv_{PVSS,A}^{SA-IND}(\lambda) = \left| \Pr[b' = b] - \frac{1}{2} \right| \quad (9)$$

Where λ is a security parameter and b and b' are guess bits.

We can consider a definition which Tian et.al used in their model [39] [40]:

Definition : a (t,n)-Threshold publicly verifiable secret sharing scheme called SA-IND (Static Adversary – IND) secure if for all probabilistic polynomial time adversaries A, (Eq. (10))

$$Adv_{PVSS,A}^{SA-IND}(\lambda) \leq negl(\kappa) \text{ for any sufficiently large } \kappa \in \mathbb{N} \quad (10)$$

Where λ is the security parameter

We can say a forward secure public key encryption scheme supporting a publicly verifiable secret sharing is called secure in SA-IND game (Static Attack) if all PPT algorithms have at most one negligible advantage in the security game. [39][34][32]

CHAPTER 6: CONCLUSION AND FUTURE WORKS

This thesis has revealed a novel security framework for public clouds called the Treasure Island Security Framework (TISF), which is based upon a DKASA Distributed file system. We have introduced DKASA and the methodology behind its proposed implementation while evaluating the security risks inherent in our approach. We believe that our proposed approach enhances both data availability and integrity while providing a higher degree of security and backup control at both the user and file level. Perhaps the most significant advantage of the DKASA cloud as proposed is the avoidance of the most common public cloud security and data availability issues; issues which have been chronicled exhaustively in both the press and the related scientific literature. Our subsequent work will seek to evaluate our claims within a pilot project which will be documented in a future paper.

As we mentioned before, such a formal treatment of key management, which act in untrusted domain with focus on running key management services in public clouds or similar domains only worked by D'Souza et.al [34]. Using FS-PKE and key management service based on individual sessions has not previously appeared. The benefits of using FS-PKE could help this new approach to act more secure and supporting of a non-interactive PVSS model, can help us to take advantage of verifying and reconstructing secret key securely on untrusted domains like public clouds. Along the same line as encryption, forward security means a break-in to current system will not affect previously encrypted information, so this feature could help us to claim our approach could work better than previously suggested systems.

Our future work will follow the current thesis, albeit with focus on the non-interactive PVSS model which was introduced briefly in the current work. We will discuss in detail how we check every step in our model and propose an implementation for our model. We already introduced all the

necessary conditions of this PVSS model specially for VERIFY() function, and as a future work we will continue working on introducing rearranged PVSS model.

REFERENCES

[1] R. Prodan and S. Ostermann, "A survey and taxonomy of infrastructure as a service and web hosting cloud providers", in 10th IEEE/ACM International Conference on Grid Computing, Alberta, Canada, 2009, pp. 17-25.

[2] K. Ren, C. Wang and Q. Wang, “Security Challenges for the Public Cloud,” IEEE Internet Computing, vol. 16, pp. 69–73, Jan. 2012.

[3] R. Iglesias, R. Nicholls, A. Travis and W. Henderson, “Private Clouds with No Silver Lining: Legal Risk in Private Cloud Services,” Digiworld Economic Journal, no. 85, pp. 125–140, 1st Q. 2012.

[4] L. Kaufman, “Data Security in the World of Cloud Computing,” IEEE Security and Privacy Journal IEEE, vol. 7, pp. 61-64, Jul. 2009.

[5] S. Subashini and V. Kavitha, “A Survey on Security Issues in Service Delivery Models of Cloud Computing,” Journal of Network and Computer Aplications, vol. 34, no. 1, pp. 1-11, Jan. 2011.

[6] J. Galante, O. Kharif and P. Alpeyev, “PlayStation security breach shows Amazon’s cloud appeal for hackers,” The Seattle Times, May, 16, 2011. [Online], Available: http://seattletimes.com/html/businesstechnology/2015071863_amazoncloudhackers17.html, [Accessed Jun. 29, 2013].

[7] B. T. Horowitz, “Microsoft Adds HIPAA Compliance Features to Azure for Cloud Health Data,” eweek.com, para. 1, Jul. 27, 2012. [Online]. Available: http://www.eweek.com/c/a/Health-Care-IT/Microsoft-Adds-HIPAA-Compliance-in-Windows-Azure-for-Cloud-Health-Data-446671/. [Accessed Jun. 29, 2013]

[8] C. Babcock, “Time to Believe in ‘Private Clouds’,” Information Week, vol. 28, no. 12, pp. 27-30, Apr. 2009.

[9] K. W. Nafi, T. S. Kar, S. A. Hoque and M. M. A. Hashem, ”A Newer User Authentication, File encryption and Distributed Server Based Cloud Computing security architecture,” International Journal of Advanced Computer Science and Applications, vol. 3, no. 10, pp. 181-186, Oct. 2012.

[10] T. Wood, A. Gerber, K. K. Ramakrishnan, P. Shenoy and J. Van der Merwe, “The case for enterprise-ready virtual private clouds,” in Proc. of the 2009 Conference on Hot Topics in Cloud Computing, San Diego, CA, 2009, pp. 4-9.

[11] D. W. Chadwick, M. Casenove and K. Siu, “My private cloud—granting federated access to cloud resources,” Journal of Cloud Computing: Advances, Systems and Applications, vol. 2, no. 1, pp. 1-16, Feb. 2013.

[12] R. Geambasu, A. A. Levy, T. Kohno, A. Krishnamurthy and H. M. Levy, “Comet: An active distributed key –value store,” in 9th USENIX Symposium on Operating Systems Design and Implementation - OSDI, British Columbia, Canada, 2010 pp. 323-336.

[13] J. Resch. “Authenticating Cloud Storage with Distributed Keys,” presented at the Storage Developer Conference (SNIA), Santa Clara, CA, 2011.

[14] A. A. Karahroudy. “Security Analysis and Framework of Cloud Computing with Parity-Based Partially Distributed File System.” M.S. thesis, East Carolina University, Greenville, USA, 2011.

[15] Maan, P. S., and Manish Sharma. "Social Engineering: A Partial Technical Attack." International Journal of Computer Science Issues (IJCSI) 9, no. 2 (2012): 557-559.

[16] B. Ives, K. R. Walsh and H. Schneider, “The domino effect of password reuse,” Communications of the ACM, vol. 47, no. 4, pp. 75-78, Apr.. 2004.

[17] Miller, Steven P., B. Clifford Neuman, Jeffrey I. Schiller, and Jermoe H. Saltzer. "Kerberos authentication and authorization system." In In Project Athena Technical Plan. 1987.

[18] S. M. Bellovin, M. Merritt, “Limitation of the Kerberos authentication system,” ACM SIGCOMM Computer Communication Review, vol. 20, no. 5, p. 119-132, Oct. 1990.

[19] D. Solo, R. Housley and W. Ford, “X. 509 public key infrastructure certificate and CRL profile,” Jan. 1999 [Online]

[20] C. Ellison and B. Schneier, “Ten risks of PKI: What you’re not being told about public key infrastructure,” Computer Security Journal , vol. 16, no. 1, pp. 1-7, Nov. 2000.

[21] A. Waqar, A. Raza, H. Abbas and M. Khurram Khan, ”A Framework for Preservation of Cloud Users’ Data Privacy using Dynamic Reconstruction of Metadata,” Journal of Network and Computer Applications, vol. 36, no. 1, pp. 235-248, Jan. 2013.

[22] S.E. Arnold, "MapReduce, Chubby and Hadoop", KM World, vol. 19, no. 10, pp. 1-18, Nov. 2010.

[23] S. Ghemawat, H. Gobioff and S. Leung, “The Google File System,” in Proc. ACM SIGOPS Operating Systems Review, Bolton Landing, NY, USA, 2003, pp. 29-43.

[24] M. Bhandarkar, "MapReduce programming with Apache Hadoop", IEEE International Symposium on Parallel & Distributed Processing, Taipei, Taiwan, 2010, p. 1.

[25] K. Shvachko, Hairong Kuang, S. Radia and R. Chansler, "The Hadoop Distributed File System",in IEEE 26th Symposium on Mass Storage Systems and Technologies, Reno, USA, 2010, pp. 1-10.

[26] T. White, Hadoop: The Definitive Guide, Sebastopol, CA: O’Reilly Media, 2009.

[27] R. L. Rivest, A. Shamir and L. Adleman, “A Method for Obtaining Ditial Signature and Public-Key Cryptosystems,” Laboratory for Computer Science, Massuchesetts Institute of Technology, Cambridge, MA, Nov. 1977.

[28] J. Daemen, V. Rijmen, “Announcing the Advanced Encryption Standard (AES),” Federal Information Processing Standards OPublication 197, Nov. 2001.

[29] INFORMS Dallas , http://meetings2.informs.org/Dallas97/, Downloaded in Feb 2014

[30] Intermediaries in Electronic Markets,

http://meetings2.informs.org/Dallas97/TALKS/MD19.html Downloaded in Feb 2014

[31] Peter Mell,Timothy Grance "The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology" Special Publication 800-145,Computer Security Division

Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, September 2011

[32] Canetti, Ran, Shai Halevi, and Jonathan Katz. "A forward-secure public-key encryption scheme." In Advances in Cryptology—Eurocrypt 2003, pp. 255-271. Springer Berlin Heidelberg, 2003.

[33] Shahbazi, Ali, et al. "A Distributed Key Based Security Framework for Private Clouds." International Journal of Advanced Computer Science & Applications 4.9 (2013).

[34] D’Souza, Roy, David Jao, Ilya Mironov, and Omkant Pandey. "Publicly verifiable secret sharing for cloud-based key management." In Progress in Cryptology–INDOCRYPT 2011, pp. 290-309. Springer Berlin Heidelberg, 2011.

[35] Goyal, Vipul, Omkant Pandey, Amit Sahai, and Brent Waters. "Attribute-based encryption for fine-grained access control of encrypted data." In Proceedings of the 13th ACM conference on Computer and communications security, pp. 89-98. ACM, 2006.

[36] Bethencourt, John, Amit Sahai, and Brent Waters. "Ciphertext-policy attribute-based encryption." In Security and Privacy, 2007. SP'07. IEEE Symposium on, pp. 321-334. IEEE, 2007.

[37] Katz, Jonathan. "Binary tree encryption: constructions and applications." In Information Security and Cryptology-ICISC 2003, pp. 1-11. Springer Berlin Heidelberg, 2004.

[38] Günther, Christoph G. "An identity-based key-exchange protocol." In Advances in Cryptology—Eurocrypt'89, pp. 29-37. Springer Berlin Heidelberg, 1990.

[39] Tian, Youliang, Changgen Peng, and Jianfeng Ma. "Publicly Verifiable Secret Sharing Schemes Using Bilinear Pairings." IJ Network Security 14, no. 3 (2012): 142-148.

[40] Tian, Youliang, Changgen Peng, Renping Zhang, and Yuling Chen. "A practical publicly verifiable secret sharing scheme based on bilinear pairing." In Anti-counterfeiting, Security and Identification, 2008. ASID 2008. 2nd International Conference on, pp. 71-75. IEEE, 2008.

[41] Farràs, Oriol, and Carles Padró. "Ideal secret sharing schemes for useful multipartite access structures." In Coding and Cryptology, pp. 99-108. Springer Berlin Heidelberg, 2011.

[42] Beimel, Amos, Oriol Farràs, and Yuval Mintz. "Secret sharing schemes for very dense graphs." In Advances in Cryptology–CRYPTO 2012, pp. 144-161. Springer Berlin Heidelberg, 2012.

[43] Waters, Brent. "Efficient identity-based encryption without random oracles." In Advances in Cryptology–EUROCRYPT 2005, pp. 114-127. Springer Berlin Heidelberg, 2005.

[44] Chatterjee, Sanjit, and Palash Sarkar. Identity-based encryption. Springer, 2011.

[45] Bethencourt, John, Amit Sahai, and Brent Waters. "Ciphertext-policy attribute-based encryption." In Security and Privacy, 2007. SP'07. IEEE Symposium on, pp. 321-334. IEEE, 2007.

[46] Waters, Brent. "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization." In Public Key Cryptography–PKC 2011, pp. 53-70. Springer Berlin Heidelberg, 2011.

[47] Anderson, Ross. "Invited lecture." In Fourth Annual Conference on Computer and Communications Security, ACM. 1997.

[48] Tsudik, Gene. "Weak forward security in mediated RSA." In Security in Communication Networks, pp. 45-54. Springer Berlin Heidelberg, 2003.

[49] Le, Zhengyi, Yi Ouyang, Yurong Xu, and Fillia Makedon. "Mobile device protection from loss and capture." In Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments, p. 41. ACM, 2008.

[50] Burmester, Mike, Vassilios Chrissikopoulos, Panayiotis Kotzanikolaou, and Emmanouil Magkos. "Strong forward security." In Trusted Information, pp. 109-121. Springer US, 2001.

[51] Stadler, Markus. "Publicly verifiable secret sharing." In Advances in Cryptology—EUROCRYPT'96, pp. 190-199. Springer Berlin Heidelberg, 1996.

[52] Chen, Yao, and Radu Sion. "On securing untrusted clouds with cryptography." In Proceedings of the 9th annual ACM workshop on Privacy in the electronic society, pp. 109-114. ACM, 2010.

[53] McKnight, Lee W., and Joseph P. Bailey. "Internet economics: When constituencies collide in cyberspace." Internet Computing, IEEE 1, no. 6 (1997): 30-37.