View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by ASU Digital Repository

Data Protection over Cloud

by

Abhijeet Srivastava

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2016 by the
Graduate Supervisory Committee:

Gail Joon Ahn, Chair
Adam Doupé
Ziming Zhao

ARIZONA STATE UNIVERSITY

May 2016

www.manaraa.com

ABSTRACT

Data protection has long been a point of contention and a vastly researched field. With the advent of technology and advances in Internet technologies, securing data has become much more challenging these days. Cloud services have become very popular. Given the ease of access and availability of the systems, it is not easy to not use cloud to store data. This however, pose a significant risk to data security as more of your data is available to a third party. Given the easy transmission and almost infinite storage of data, securing one's sensitive information has become a major challenge.

Cloud service providers may not be trusted completely with your data. It is not very uncommon to snoop over the data for finding interesting patterns to generate ad revenue or divulge your information to a third party, e.g. government and law enforcing agencies. For enterprises who use cloud service, it pose a risk for their intellectual property and business secrets. With more and more employees using cloud for their day to day work, business now face a risk of losing or leaking out information.

In this thesis, I have focused on ways to protect data and information over cloud- a third party not authorized to use your data, all this while still utilizing cloud services for transfer and availability of data. This research proposes an alternative to an on-premise secure infrastructure giving flexibility to user for protecting the data and control over it. The project uses cryptography to protect data and create a secure architecture for secret key migration in order to decrypt the data securely for the intended recipient. It utilizes Intel's Identity protection technology (IPT) with Public key Infrastructure (PKI) for providing a hardened security which gives an added advantage over other existing solutions.

www.manaraa.com

DEDICATION

To my parents,

***Rakesh Kumar Srivastava & Manju Srivastava**,*

It would not have been possible,

without your support and sacrifices.

www.manaraa.com

ACKNOWLEDGMENTS

I would like to take this opportunity to express my deep grteatitude to my advisor and mentor, Dr. Gail Joon Ahn. This thesis is the outcome of your help, insight and specially your boundless patience throughout this process. Your help and support motivated me and kept me going. You are truly a great mentor and a gem of a person. This is possible because of your unending support and invaluable feedback. Thanks a ton! I would also like to thank Professor Raghu, who taught me how to tackle problems when you are stuck and gave me valuable advice which I will cherish throughout my career. I would also like to thank Professor Adam Doupé and Ziming Zhao for being on my committee and giving their feedback on my thesis.

I would like to thank my father, Rakesh Kumar Srivastava and my mother, Manju Srivastava, for their love and support which made me what I am today. Thank you mom and dad!

I would like to thank my team at Intel for providing me any help I needed from time to time. Thank you Hormuzd for believing in me and providing your guidance. Thank you David and Khaled for all your support, you were always available for any help and have listened to me patiently. Special thanks to my manager, Yasser Rasheed for making it possible.

www.manaraa.com

TABLE OF CONTENTS

www.manaraa.com

CHAPTER Page

www.manaraa.com

LIST OF TABLES

www.manaraa.com

LIST OF FIGURES

www.manaraa.com

Chapter 1

INTRODUCTION

1.1 Motivation

Data protection has long been a point of contention and a vastly researched field. It has been a need before the invention of first cipher in the history. Civilizations used various techniques to transmit message securely. With time the medium has changed but the requirement is still as critical as it was before. Today, in the digital age, we not only create data at a superfluous rate we also store, share and transmit a lot of data. According to Alphabet CEO Eric Schmidt, each day we generate as much data as we did from the start of civilization up to 2003 [70]. This gives an idea about the scale of information we produce which is accessible to people. A major portion of which could be sensitive or private. Once a piece of information is digital it is not easy to contain it to yourself or the intended audience. Data over the Internet is accessible to people whom we do not know about and are snooping over our information. With the advent of large server farms and infinite data storage, storing multiple copies of your data at many location is not very uncommon. Though it has its benefits, it raises concerns among users and businesses. Documents which are not meant to be public, remain susceptible to attacks when there are multiple copies of it.

In recent years we have witnessed an explosive growth in cloud related technologies where users can access the services on a rental or temporary basis. Cloud services have been rising in popularity significantly as it provides a cost effective and highly available model of fulfilling the computing needs for a personal user or an enterprise.

www.manaraa.com

It is more appealing, given the ease of access and availability of the systems but they do pose a significant risk to your data.

However, usage of cloud storage systems increases the risk many fold as there are not just multiple copies now- to increase the availability of your data from anywhere quickly, there is also a third party involved which has control over your data. Cloud service providers may not be trusted completely with your data. It is not very uncommon to snoop over the data for finding interesting patterns to generate ad revenue, or divulging your information to a third party e.g. government and law enforcing agencies. In a recent report by electronic frontier education [66], out of the 24 companies evaluated, only fifteen even notified the users about disclosing their data to government agencies. An interesting thing to note is that this report is an improvement from past years. Surely, Silicon Valley is taking up the privacy issue seriously but for more protection user should step up and take the task of securing the data in their hands.

1.2 Problem Statement

Cloud operates in three service models: software as service (SaaS), platform as service (PaaS) and infrastructure as service (IaaS). There are various security concerns regarding cloud usage, e.g. privacy, user control, etc. which are discussed in detail in [73] and briefly later in this manuscript. In a traditional on-premise application, data is stored within the enterprise boundary. However, with the usage of cloud service models, data is sent outside this secure periphery for computing or storage. This poses a risk for the enterprise as they do not have control over the data. This work focuses on data storage over the cloud. Using cloud services for file storage gives a lot of benefit but comes at an added risk. Service providers operate on multiple geographies and come under different laws, and under certain regulations they might

www.manaraa.com

even have to disclose the documents they have stored on the server. This - in case, cloud is trustworthy and has no motivation to access your data. For large enterprises, this may not be a viable way since company documents may contain intellectual property and business secrets. There are multiple concerns with cloud data storage as discussed in [44], [63], [62]. These include, but not limited to:

1. Privacy
2. Lack of user control
3. Security
4. Trust issues
5. Legal aspects

This thesis targets to resolve the issues faced in data storage on the cloud.

1.3 Existing Solution

Traditionally, security for an enterprise meant accessing the data in a restricted environment which can be controlled or regulated by IT department. Thus firewalls, intrusion defense systems were able to resolve the security issues. However, computing in cloud does not let your traditional security measures be as effective as they were before. One needs to revamp the security measures for cloud computing. There have been efforts to address the issues arising due to usage of cloud, but most of them focus on on-premise cloud which is expensive and difficult to maintain and only large companies can afford it. Hwang and Li [56] propose a method to use a trust-overlay network across multiple data centers to establish trust between the business and the cloud. This however assumes cloud is not an adversary and can not be forced/motivated to leak the data it stores. Shen and Tong, in their work [69]

www.manaraa.com

realized the need of trusted computing for a secure cloud storage. Chow, Golle, et al. [50] in their survey also found trusted computing for cloud data protection to be a better alternative.

There are solutions available to mitigate the concerns discussed before. One of them is encrypted cloud, where the data is stored in an encrypted form. Products like CipherCLoud [17], Boxcryptor [14], BetterCloud [13] provide these facilities. This enables the user to store the document in such a way that only the intended recipient or the owner be able to utilize the data. However, the key to the data is not controlled by the user/enterprise. For a enterprise, this pose a problem when an employee leaves the company and it has to reach out to cloud service provider (CSP) to let them have access to employees documents, making them dependent on CSPs.

There have been efforts in giving the security of users data in user's hand but most of them have the problem with the storage of key. A technique to provide protection over cloud is discussed in [74] and provides the protection from cloud. However it does not addresses the lack of trusted computing and is prone to attacks. Products like Boxcryptor [14] also provide similar functionality but store the key on the system and are susceptible to a malware intrusion. Other solutions, e.g BetterCloud [13], CipherCloud [17], which aim at protecting data over cloud but lack trusted computing integration are much prone to attacks than a trusted computing based solution.

This research at Intel overcomes these limitations by making use of Intel's firmware based key encryption where the key is exposed only in a secure environment. This hardens the security of the system and almost nullifies malware or any software based attacks. Intel Identity Protection Technology (Intel IPT) [7] with Public key infrastructure (PKI) is a second factor authentication for business and web services that validates when a legitimate user, not malware, is logging in from a trusted PC. This technology is available on 3rd generation and higher Intel Core vPro processors [7].

www.manaraa.com

It utilizes Intel Management Engine (Intel ME) to provide a trusted measure for authentication. Intel ME lets one save certificates on a system which can be used to authenticate and provide hardened security features. The research also proposes a scheme to keep the key to the document under user/enterprise control and make them independent of the service provider.

1.4 Overview

This research focuses on security challenges faced by enterprises and users in general by utilizing cloud for storing their data. For enterprises, the traditional model of on-premise security has not been very effective to prevent data thefts [6]. With increased usage of cloud services by employees, this risk has become more widespread and there is a need to protect documents at a more granular level instead of the traditional *barrier* security.

This work focuses on analyzing the existing research for the above problems along with the concerns regarding the ownership and control of the secured data. It aims at giving control of security in the hands of enterprises instead of trusting a third party, in a way which is feasible and is an improvement over the existing solutions. The research comes up with a way to securely protect the document and have the security in control of the enterprise by giving them control over their key. This research focuses on two important aspects:

1. Protecting data on the disk

   Researching on ways to prevent the data stored on the local system from being accessed by the CSP [65]. The aim is to have control and secure the data before it is accessible to the CSP on local system. A brief overview is depicted in 1.1. The user operations on file are captured by a file system filter diver [40] which

www.manaraa.com

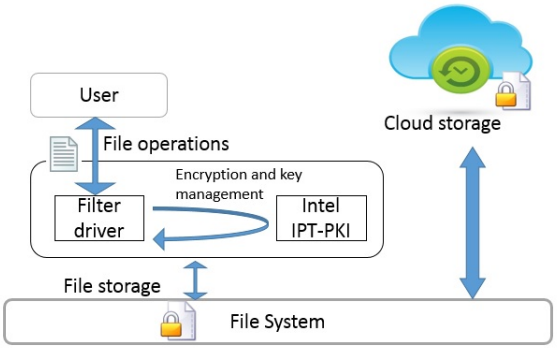

Figure 1.1: Protecting data on the disk

interacts with IPT-PKI for key management and encrypts the file at run-time and stores in the hard disk. This prevents the CSP to intercept the data in plain text and only uploads encrypted content to the cloud.

2. Key control and secure migration

   Having the data restricted from cloud is not enough since the mobility of data is required. This aspect of the research aims to transfer encrypted data to cloud and keep the key to secured documents in enterprise's control. A central and highly secured server is required for key management which is responsible for user authentication and exporting or importing the keys.

Figure 1.2 gives an overview of the approach that has been used to protect the data from cloud and key migration to provide the needed mobility. My approach utilizes Intel's technology for key protection on the local system, providing a hardened security feature. This provides an improvement over other previous work which lacked hardware based security for user authentication and malware protection.

The structure of the document is as follows. Chapter 2 gives an overview of cloud computing and why security is an issue. It also present statistics about enterprises being at increased risk due to cloud services. Chapter 3 discuss the role of cryptogra-

www.manaraa.com

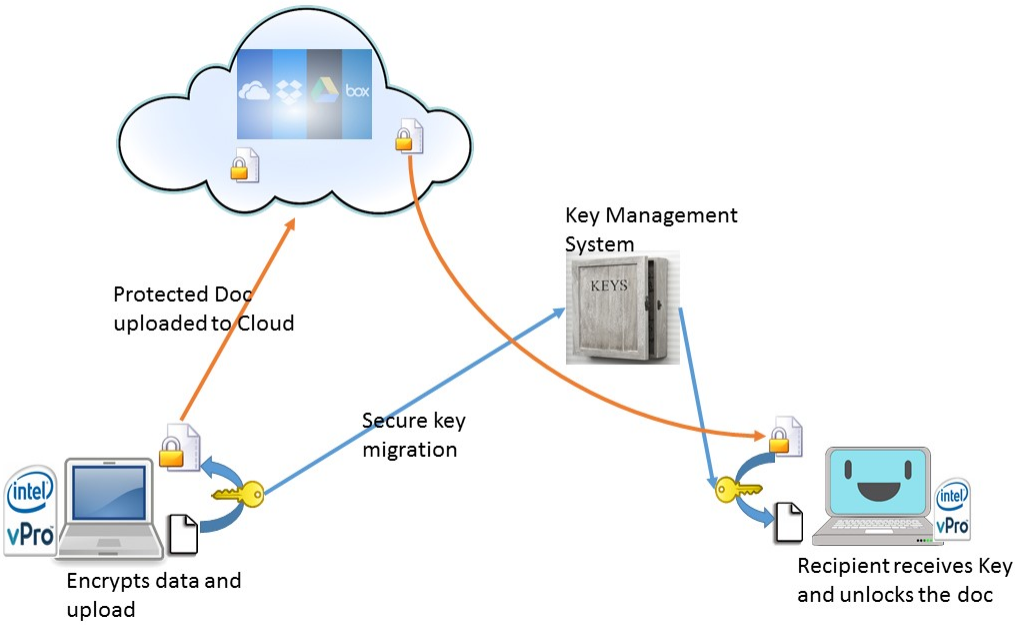

Figure 1.2: Overview of Data Protection approach

phy and evaluate cryptography algorithms which are suited for the project. Chapter 4 discuss the realization of the architecture and data protection approach. 5 discuss the implementation detail and evaluation of the proof of concept developed. The thesis concludes with discussion on further work in the area in chapter 6.

www.manaraa.com

Chapter 2

CLOUD COMPUTING AND SECURITY

As discussed in Chapter 1, data over cloud raises many security concerns and has implications which may not be apparent immediately. With all the benefits and flexibility a cloud computing services provide, it comes with its own risk and privacy. In this chapter, I give a brief overview of cloud service and describe the security risks faced by an individual or organization by storing data over cloud and its implications. I will also present how common the attacks on data have become in recent times and the impact malicious attackers have on the community as a whole.

2.1 Cloud Computing

This section gives a brief overview of cloud service model and how it has changed the industry and way people use and process digital content. Cloud computing is model which lets a user access various shared services on demand and with minimal management efforts. Concept of cloud service is not a new one, it has been in use since the early origins of computing. The idea of cloud computing was supposedly introduced by J. C. R. Licklider while developing ARPANET (Advanced Research Projects Agency Network) in 1969, to connect people and data anytime anywhere [29]. The earlier versions of computer network had users access the central computer by connecting via terminals, small hardware with limited or no computing capabilities. These early terminals did not do much other than send and receive characters from a mainframe or centralized computer as in a point to point network [71]. Over the period of time these terminals got smarter as they were able to do more processing at the terminals and most of the computing was done at the local site instead of a central

www.manaraa.com

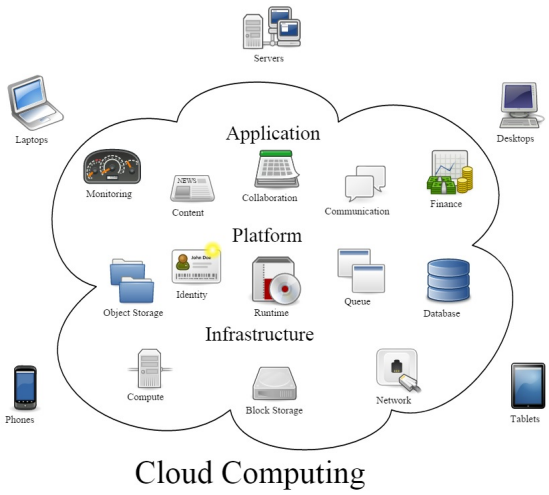

Figure 2.1: Cloud Computing [57]

server or computer. As time progressed and with advent of PC, individual computers became the de-facto and widely used across industry. With further revolutions in IT, networking and Internet, a new paradigm of computing has evolved, Cloud Computing. This model does not limit a workstation capabilities with its hardware, instead it allows to use the services provided by a shared and much better pool of resources. It provides users and enterprises with various capabilities to store and process data in third-party data center and utilize large shared resources [53]. The figure 2.1 shows a high level overview of cloud architecture. Traditionally, cloud operates in the below service models.

1. Software as a service (SaaS): It is a software delivery model in which the software is centrally hosted and is licensed on a subscription basis.

2. Platform as a service (PaaS): This version of cloud allows users to run their application using the cloud service providers platform. e.g. develop, run and manage web-based application without developing and maintaining the infrastructure required for the software.

www.manaraa.com

3. Infrastructure as a service (IaaS): This model provides the infrastructure required for computing like operating systems, environment etc. e.g. Amazon Web service (AWS), Windows Azure, Google Compute Engine.

2.1.1 Data Storage over Cloud

In order to understand why cloud pose a concerns for your data, it is necessary to understand how it stores the data and what enables it to let the data be available to you anytime and anywhere. The capabilities which allow this flexibility in turn pose risk for the users. Cloud is a made up of many distributed databases but still allows to be treated as one. Among many feature of cloud it is noteworthy to realize that it maintains high tolerance through redundancy and distribution of data. Data which is stored at a server is often duplicated and kept at multiple locations which can be located across geographies coming under different laws and regulations. This can be risky in case of a government agency asking the CSP to reveal the data they have stored in the servers.

My research intends to mitigate this problem by encrypting the document so that even if CSP wants to reveal the data at their server, it will be of no use without the key to decrypt the document which is in control of users/ document owners.

Cloud Data Storage tools

There are many products available in market which provide on line storage for your data. This section will give a brief survey of the available products.

1. Dropbox: It is file hosting service operated by Dropbox Inc, headquartered in San Fransisco, California [25]. It allows user to create folders in their local system which it synchronizes to the server. File are accessible from Dropbox website or mobile application.

www.manaraa.com

2. Google Drive: It is cloud based file storage and syncing service [28]. It also lets one create, share and collaborate on documents over the Internet. It is available for free unless the data size is less than 15GB.

3. SkyDrive / OneDrive : It is a file hosting service offered by Microsoft which lets one create and store file and later access by web or mobile [10].

2.1.2 Security Concerns

With user data stored on a third party resource many security issues arise. Below are short description of issues as enumerated in (Benameur, et al)[44], [63], [62], [43].

Privacy

Privacy at a coarse level refer to users control of their data and reserving the right to expose it to others. For organizations, it entails the way they store and manage personal identifiable information. Public cloud access is the most common way of accessing the cloud as service (which can be accessed over Internet and intranet) but relying on the cloud service provider to handle the confidentiality of the data raises many concerns.

Lack of User control

While accessing cloud service it is not expected from service provider to cater to users demand of restricting storage of their data and duplicating it. Data is stored over machines where users do not have any control and places where the laws can be different from the place of origin.

www.manaraa.com

Unauthorized Secondary Usage

Data over cloud is a great source of revenue of the service provider as it provides a resource for mining interesting patterns and target ads. Moreover there is no guarantee if the service provider goes bankrupt and the sells the data to gain more revenue. These unauthorized data usage pose a great risk to the user and organizations using the cloud service.

Data Proliferation and Trans border Data Flow

Data in cloud involves multi-parties and multi storage sites. User, who is the data owner has no control over the sites and parties involved in the processing of his/her data. When data in cloud is moved across sites falling under different legal jurisdiction increases risk and factor and legal complexities [43].

Security Issues

Traditional security measures considered a secured perimeter with a trusted boundary-within which information is stored and processed, to be secure. But this has changed a lot within few years with increased access to cloud and distributed computing. People and data have been more dynamic and information exchange has been increasing. With data being freely shared and distributed, It poses danger of information leak and it is more severe for enterprises who use cloud services. It endangers their Intellectual property and business secrets.

Traditionally, enterprises enforce IT policies and security constraints to restrict the flow of data outside their organization. With more and more employees using cloud for their day to day work, business now face a risk of losing or leaking out

www.manaraa.com

information. A document once uploaded over cloud may not fall under the owners jurisdiction and is susceptible for copying or any malicious usage.

Bring your own device (BYOD) [15] is becoming very popular these days among employees, as it gives the much needed flexibility to the employees and increase productivity. It is a growing trend among the employees to use their personal devices for work purposes. According to a survey done by security intelligence which provides analysis and insight for information security professionals, almost 60 percent of the enterprises allow their employees to bring their own devices to work [3]. This shows how popular usage of non-IT systems is. This increases the risk of information leakage manifold as now not all the personal laptops and devices are enabled with IT policies of the organization and not as secure as an on-premise computer. Another report by Gartner [1] predicts that by 2017, almost half of the employers would require employees to use their devices for work purposes.

BYOD does increase productivity and helps get things done virtually from anywhere, but a certain solution is needed to compensate the risk factors involved.

	Smart Phone	**Tablet PC**	**Laptop**	**Percent of cases**
Technology giant	35.7%	28.6%	35.7%	100%
Educational Institution	34.8%	34.8%	30.4%	100%
Financial Organization	42.9%	25.7%	31.4%	100%
Total average	37.8%	29.7%	32.5%	

Table 2.1: BYOD devices usage frequencies in the organizations [5]

Trust Issues

With the presence of multiple vendors offering distributed computing and cloud storage, there has been a race to get to cloud. Business now have many options to choose from. However, the dynamically changing industry and the sensitivity of data may

www.manaraa.com

be inhibiting factors against choosing for cloud. Trust is a main concern with many industries, mainly health and financial due to the sensitivity of the data being stored. Since user does not have control over his/her data over cloud, they may not completely trust the cloud service provider specially if the data is related to their health and finances.

Recent reports in media hint at government agencies like NSA being able to mine the data stored over cloud without any prior permission. NSA PRISM program in collaboration with large tech companies intends to do so. Media articles even suggest that Microsoft gave NSA permission to record Skype calls and go mine user data store over cloud [2].

This research intends to mitigate this concern as now user has complete control over his/her data. Since the data is encrypted and stored on the cloud, CSP can not leak info about user to any other party even if they want to. However, it is still susceptible to techniques which aim at finding patterns/ keywords in encrypted content [46].

Legal Aspects

Since data in cloud is stored at multiple places across geographies, laws of the land could be different and may not be strict enough to protect user data. Moreover, cloud service provider always get request from government and securities agencies to disclose the data they have stored and are bound by law to disclose it. The USA Patriot Act signed by Bush on Oct 26, 2001 gives the US government unfettered ability to obtain access to data stored outside the United States by U.S. cloud service providers or their foreign subsidiaries[9].

The presented thesis intends to solve few of the above mentioned problems. Using an end to end encryption scheme, with separate control of key resolves below concerns:

www.manaraa.com

1. Privacy: With end to end encryption of data, the risk of leaking data by CSP is drastically reduced. Since it is very hard to make sense of encrypted data without it's key, User may not worry about the content even if the encrypted document is available to an adversary.

2. Lack of User Control: This research proposes methods by which users will have the ability to share the keys to other users. This gives user control over their data while still maintaining the privacy and secrecy.

3. Unauthorized Secondary Usage: Since the data is encrypted in the proposed approach, it is difficult to use it for data mining or web scraping. However it is susceptible to techniques which aim to search for patterns in encrypted contents [46].

4. Data Proliferation and Trans border Data Flow: This thesis present the approach which resolves the above problem. Even if data is stored at multiple places and under various government regulations it is of no use even if exposed to authorities at the server.

5. Trust Issues: In the proposed approach, the key control is in users/enterprise hand and not in control of CSP. This makes the infrastructure more trustworthy as a third-party, the CSP, is not able to expose or leak out the sensitive data. The users have control over who can access the data.

6. Security Issues: With the usage of Intel IPT-PKI [68], the proposed technique is resistant to any malware and works only a trusted machine.

www.manaraa.com

2.2 Recent Attacks and Concerns

This section will briefly enlist some recent attacks to give an idea of the impact of data breaches on corporation. The intent of this section is not to enlist every attack but only to give an idea of the importance of protecting data over cloud.

The recent cyber-attack on Sony Pictures Entertainment [38], allegedly due to the release of the motion picture "The Interview" can be a lesson for businesses to put in more efforts in protecting the data in cloud [52]. Analysts are predicting that the data breach could have cost Sony as much as 100 million USD not withstanding the lost goodwill and brand image of the company [52]. Recently IDC, International data center, predicted that by 2020 more than 1.5 billion people will be affected by data breaches, that is almost a quarter of world population [72].

There is a growing need for a hardened security feature for enterprises which today face attacks from hackers. The emphasis of this thesis is to provide a hardened security by using Intel's latest technology which would help prevent data breaches. Patrick Moorhead, an industry analyst with Moor Insights and Strategy noted that "If Sony and Target [8] had that hardware-based, multi-factor authentication, they wouldn't have been hacked successfully" [48] .

2.3 Cloud Security Solution

Information Security is not very black and white to implement. The data one wants to protect may not always be static or restricted to one person. Information is created to be shared among many stake holders. Information is not monolithic, various access levels for users are required according to their privilege levels. Information right management solves this to some extent where user have different permissions for any given document. However, this does not protects the data from the cloud.

www.manaraa.com

Given the popularity and various advantages of cloud services usage of cloud is very difficult to avoid.

A general security solution in enterprise is either very restrictive or very open. Traditional security measures assumed that information consumption in a secured and bounded environment is secure enough. But both the solutions have their faults, the complete and unshared security limits the users from being productive and makes it hard to share even with an intended recipient. Using an enterprise *cordoned off* environment does not restrict users to upload on cloud and is susceptible to cloud services.

2.3.1 Cloud Security Solution Requirement

A Cloud security solution must have the following features which would solve the purpose of security and also allow the flexibility of cloud.

1. **Selective Security**

   It should allow Users to mark any folder as secure. Users must have flexibility to protect specific documents they want. A complete system-level security hampers the performance and is very user intrusive which may lead to users finding a way around and defeating the purpose of information protection.

2. **Cloud Access**

   Even though uploading data on cloud in plain text is risky, the requirement here is to allow users to upload the data to cloud in a secure fashion. The solution should meet the corporate IT standards and cloud provider should have no visibility into the data.

3. **Sharing**

   The users should be able to share the documents with any authorized users. To

www.manaraa.com

provide a hassle free sharing, the users should be able to use a single click UX to share data.

2.3.2 Intel IPT with PKI

Intel Identity Protection Technology (IPT) enables users with hardware based security and authentication mechanism well suited for web and enterprise usage. It is a robust, convenient way to deter identity thefts. With growing software based attacks, this technology provides a hardened firmware based protection. This makes it resilient to any malware running on the infected system and allow authentication only on a trusted system. Some of its features are mentioned below [68]:

1. **Strong Protection**

   This technology inherently benefits from the comprehensive hardening measures implemented in Intel Management Engine (ME). It is rooted in tamper-resistant hardware and provides a much greater security than any software based security system.

2. **Low Cost and Transparency**

   The security feature do not require any specific or extra hardware and is cost effective. Once installed, the user need not provide an extra set of credential or token for a second factor authentication. He can simply provide his credential for any website he wants to access.

3. **Multiple token in one device**

   With many enterprises providing users with a separate token for additional security measure, it is susceptible to loss of it. Using Intel's technology, the token is inbuilt in the hardware and it removes carrying any separate tokens. Furthermore, more than one token be built into the system.

www.manaraa.com

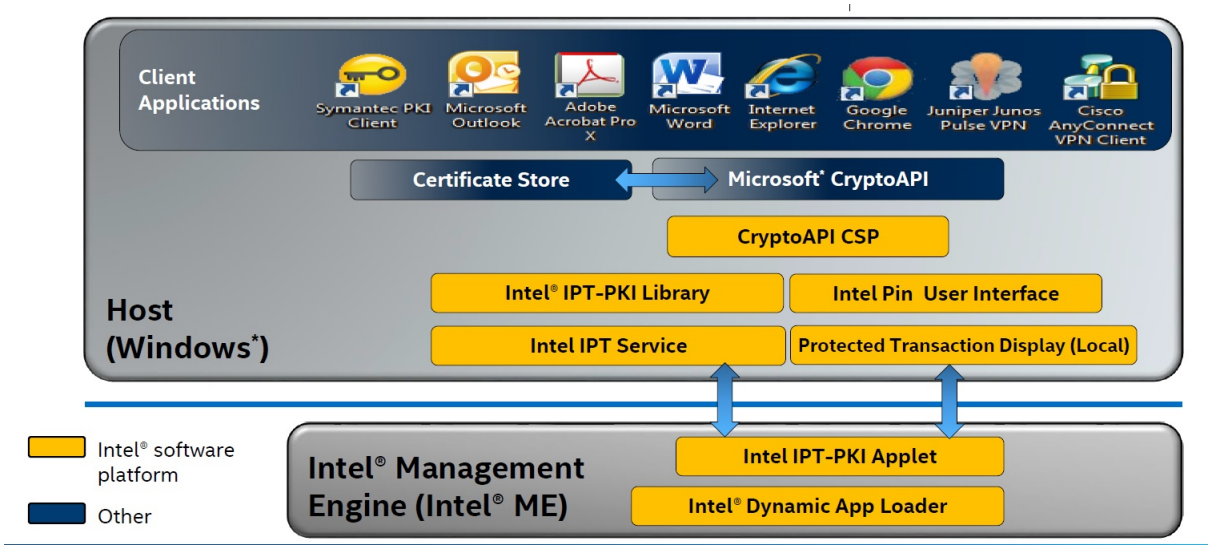

Figure 2.2: Intel IPT v1 architecture [59]

Below are mechanisms how Intel IPT works [68]:

1. Intel IPT with multi factor authentication(MFA)

   Intel IPT with MFA is a connected framework which provides a seamless solution for policy based identity and access management. It integrates well within an IT infrastructure and gives a firmware based authentication of user, platform and network [68]. IPT integrates with windows applications using Microsoft crypt APIs. It is similar to how smart card are usually integrated in windows [67].

2. Intel IPT with one-time password (OTP)

   Intel IP uses OTP tokens to strengthen the network and web site access, using a second factor authentication. When a user visits a website which enables Intel IPT, it can bind the user account with the device using the OPT token generated in the firmware. This gives a cheaper and much stronger two-factor authentication.

www.manaraa.com

3. Intel IPT with public key infrastructure (PKI)

   It also includes support for Public Key infrastructure, where Rivest-Shamir-Adleman (RSA) [37] key pairs and certificates are generated in the embedded security processor. RSA is a cryptosystem which is widely used for secure communications. This helps authenticate user to a device using OS login and device to a network via VPN. This avoids the cost of traditional smart card reader or specially ordered PCs.

4. Intel IPT with protected transaction display (PTD)

   Intel IPT can display information and receive input from user using the embedded security processor. This prevents a malware scraping for inputs since it requires a human presence to display or accept data.

2.3.3 Data Protection Using Intel IPT with PKI

Use of Intel's Technology provides an added level of security for data protection and can be used by the enterprises to protect their sensitive information from being leaked over to cloud. This research takes advantage of this technology to provide a data security measure for any individual or enterprise. The *Security Key* used to encrypt and decrypt data is kept securely in the firmware and any interaction or authentication is done in the separate security processor. Figure 2.3 depicts the flow of the key which is generated in firmware and how it interacts with the Operating System which will subsequently be used by our application. The key is generated in hardware using Intel Management Engine (ME), encrypted with a platform binding key (PBK) [47] and passed back to Windows. PBK is unique for each platform using Intel IPT with PKI. Windows then store the encrypted key in user library for storage.

www.manaraa.com

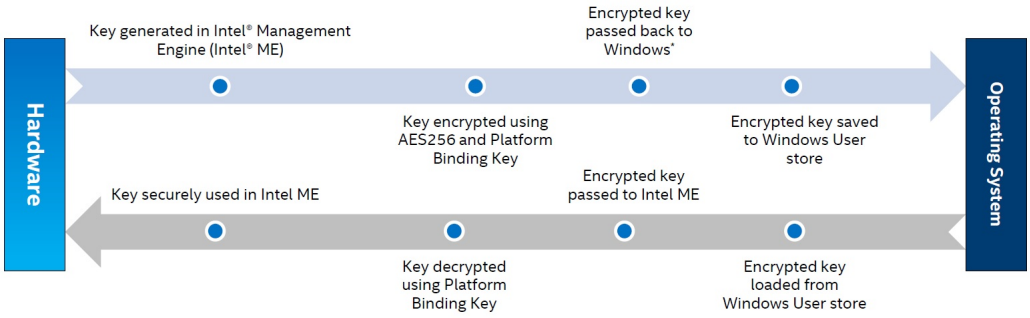

Figure 2.3: IPT Key usage and storage [59]

For using the key, the encrypted key is passed to Intel ME which is decrypted using PBK and key is securely used in ME for authentication or any other purpose.

For this thesis we leveraged IPT with PKI version 3.0 which has the feature of secure import and export and gives us the capability to trust that we are running on an authentic system because of the following features:

1. Secure Import for PKI key-pair/certificate

   Allows to scale Intel IPT by protecting non self generated certificates. Although these must have been generated by Intel IPT-PKI on another system.

2. Hardware based key attestation.

   It uses Enhanced Private ID (EPID), a symmetric key embedded into Intel Hardware at manufacturing time. This provides extra protection from man in the middle attacks.

3. This version of IPT-PKI is well suited for enterprise usages as it can be used across multiple devices and used in Data Protection.

www.manaraa.com

Chapter 3

ELEMENTARY CRYPTOGRAPHY

To protect data from any unauthorized usage, cryptography comes in handy. Cryptography means encoding a meaningful data such that only the intended recipient and the sender can make sense out of it, i.e. decode it. This chapter will give a brief of some important concepts about cryptography which have been used in my thesis. The chapter starts with giving a description about encryption and decryption, the concept of keys, and authentication. It also talk about the feasibility of encryption algorithms which have been used to solve the problem of data protection.

3.1 Cryptography

Cryptography- art of secret writing is an effective method of controlling against many kinds of security threats. Well disguised information can not be read or modified easily, which protects it form unwanted interference [49]. Cryptography is rooted in higher mathematics and number theory but a deep knowledge is not required to be able to use cryptography.

Let us consider a scenario of message passing from Alice to Bob. Alice and Bob are commonly used names in describing cryptographic examples. Communication is done over a transmission channel and a third person, Chuck, an adversary wants to hinder the message communication. Chuck may try the following [49]:

1. Block it, by preventing it reaching Bob, thereby affecting the availability of the message

www.manaraa.com

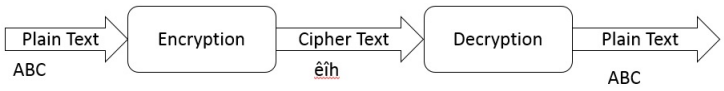

Figure 3.1: Encryption and Decryption

2. Intercept it, by reading or listening to the message, thereby affecting the confidentiality of the message

3. Modify it, by seizing the message and changing it in some way, affecting the message's integrity

4. Fabricate an authentic-looking message, arranging for it to be delivered as if it came from another person, thereby also affecting the integrity of the message

Fortunately cryptography can help address all these points. Encryption is the process of changing a message such that it not meaningful or is not obvious. Decryption is the reverse of encryption and converts the meaningless data to its original meaningful form. Figure 3.1 shows the encryption and decryption for a plain text. Plain text is the useful data which one wants to secure by converting it to a random - not useful gibberish called cipher-text, which in-turn is converted to plain text using the decryption process.

An example of encryption would a simple scheme of substituting the plain text data with seemingly random characters. The aim of the deign is such that only the person who knows how this scheme replaced the characters would be able to regenerate the original message.

www.manaraa.com

3.1.1 Substitution Cipher

It is one of the simplest scheme where a character is replaced using a substitution table. A historic and simplest example could be **Ceasar Cipher** where each character is replaced by shifting it a fixed number of places in the alphabet series. It was used by Julius Ceasar where he used the shift as 3, So the letter 'A' would become 'D'.

Plaintext A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Ciphertext d e f g h i j k l m n o p q r s t u v w x y z a b c

E.g. Using this scheme the message 'APPROVE' would be encoded as 'dssuyh'

Analyzing the Ceasar Cipher, one would see that it is easy to remember and it worked well in the time it was invented. It served the purpose of converting plain text data to seemingly randomized data which would be easily decoded by the person knowing the scheme. However, it also had a weakness that a small piece of encoded data would reveal the encoding of the entire data. This, is not a desirable property in any encryption algorithm.

3.1.2 Symmetric Encryption

Symmetric Encryption is the oldest and best-known technique. In this scheme of encryption a plain text is encoded using a *Secret key*. This key can be a combination of letters, alphabet or just any string of random characters. Using this key, encryption can be done using any scheme or algorithm. The important characteristic of symmetric encryption is that the same key is used to convert the encoded message back to plain text in decryption stage. Usually, the algorithm for decryption of the crypt text (encoded message) is almost similar to that of the encryption. E.g. In Caesar cipher, the encryption is done by replacing the characters 3 letters *later* in the alphabet. Similar to the encoding technique, the message can be decoded by

www.manaraa.com

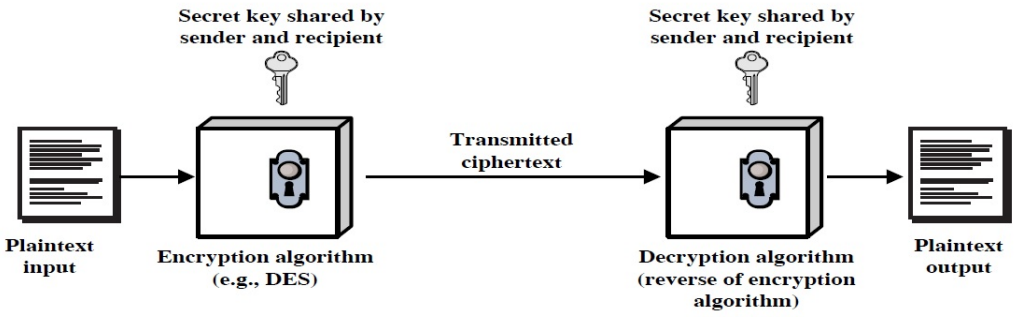

Figure 3.2: Simple Model of Symmetric Encryption [61]

replacing the characters 3 letter *earlier* in the alphabet. Figure 3.2 shows a simple model for symmetric encryption.

The Symmetric key Encryption has five components:

1. **Plain text** The original message which needs to be encoded

2. **Encryption Algorithm** The Encryption Algorithm converts the plain text to random data by performing various permutations and substitutions depending on the type of algorithm used.

3. **Secret Key** The key which is used to encrypt and decrypt the data.

4. **Cipher Text** The encoded message generated from the plain text using the secret key.

5. **Decryption Algorithm** The algorithm to decode the encoded data, usually very similar to encryption algorithm.

Symmetric encryption relies on the secrecy of the key which is used to encrypt the data. e.g. If Alice wants to communicate with Bob using a public channel of communication, Alice can use symmetric encryption to encrypt the data and send it over to Bob. However, the key must be securely passed to Bob and should not be

www.manaraa.com

compromised. Bob on receiving the encoded data can decrypt using the key given by Alice. A point worth noting here is that one can argue that if key can be securely transferred then why can't the data itself be transmitted securely. This is so because usually the actual data is much bigger in size than the key which is used to encode it. Securely transmitting a small data (key) is easier than transmitting the actual larger data. This is the reason why an almost perfect encryption scheme called 'One time pad' is not practical. **One Time Pad** is an encryption scheme where a plain text is encrypted using a key of same length (*One time pad*). If used correctly, this gives a perfect cipher which can not be cracked. This encryption scheme did not get much popular due to the impracticality of securely transmitting the key which is as large as the secret message it encodes.

3.1.3 Asymmetric Encryption

So far we have discussed about how to mangle data so that it is not easy for any one to make sense from it. But for the intended recipient it is easy to get the data back using a secret key. The problem with symmetric encryption is that the key should be transmitted securely and that there should be trust between the two parties, e.g., Let's say Alice and Bob want to securely transmit some data. There are two main concerns which may arise:

1. **Secure transfer of key**

   Alice needs to securely transfer the key to the she wants to communicate with. And in case there are n people with whom Alice wants to communicate, she has to give each person a new secret key. So that one person may not know other person's message.

2. **Key Proliferation**

www.manaraa.com

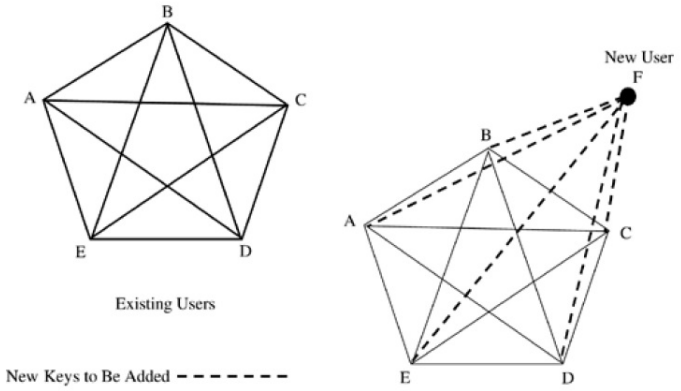

Figure 3.3: Key Proliferation [49]

As mentioned above Alice needs to maintain n keys in case she wants to communicate with n people. In general, an *n-user* system requires *n*(n-1)/2* keys and also each key must be specific to each user. As the number of user increase the number of key increase rapidly. Figure 3.3 shows how addition of a single new user affects the existing key infrastructure.

3. **Repudiation**

   In case there are trust issues, Alice may deny sharing any key with Bob. So Bob can not prove the data he decrypted is actually what Alice really sent.

Asymmetric encryption or public-key cryptography resolves both the issues. The concept originated with Diffie and Hellman's breakthrough at Stanford in 1976 [58]. Their major contribution to cryptography was the idea that keys could come in pair and it would be impossible to generate one key from the other. This comes in handy for transmitting encrypted data over an insecure channel and removes the requirement of transmitting the secret key securely as required in symmetric encryption. Below are the ingredients of asymmetric encryption:

1. **Plain text**

   The actual data which is fed to the encryption algorithm to be encoded

www.manaraa.com

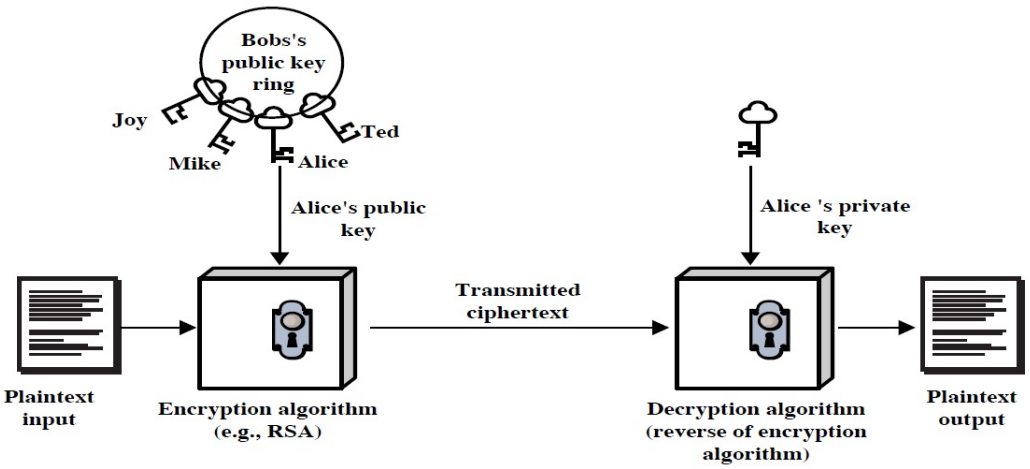

Figure 3.4: Asymmetric Encryption [61]

2. **Encryption Algorithm**

   Algorithm performing the randomization or encoding of plain text.

3. **Public and Private Key**

   This is a stark feature of public key cryptography, the pair of keys work in tandem for encryption and decryption. One key is kept private and another is available freely to anybody, hence they are referred to as private and public key. Although any of the key can be public or private. One of the key is used to perform encryption and the other key is used to perform the decryption.

4. **Cipher Text**

   The encoded or scrambled message after the encryption.

5. **Decryption Algorithm**

   The algorithm converts the cipher text to plain text.

www.manaraa.com

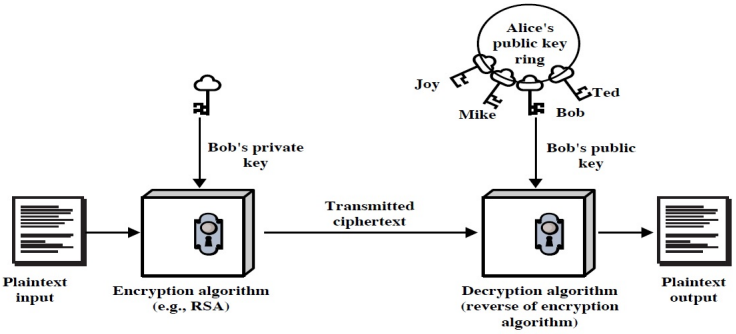

Figure 3.5: Asymmetric Authentication [61]

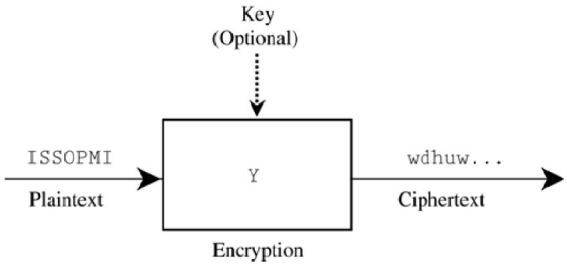

Figure 3.6: Stream Encryption [49]

3.1.4 Stream and Block Cipher

Stream ciphers, as the name suggest transform the data symbol by symbol. i.e. one plain text character is encoded to a cipher text character. The encoding depends on the input symbol, encryption key and the algorithm. Due to this property of stream ciphers, an error in the encryption of one character, such as missing a character, can mess up the encryption of further characters in the stream. These errors can be recognized as the plain text will be recoverable up to a certain point and then the error can be fixed. However this makes the encryption susceptible to cryptanalysis [18].

This problem is addressed by using another form of encryption called, **Block Cipher**. Block Cipher encrypts a group of plain text symbols as one block. It works

www.manaraa.com

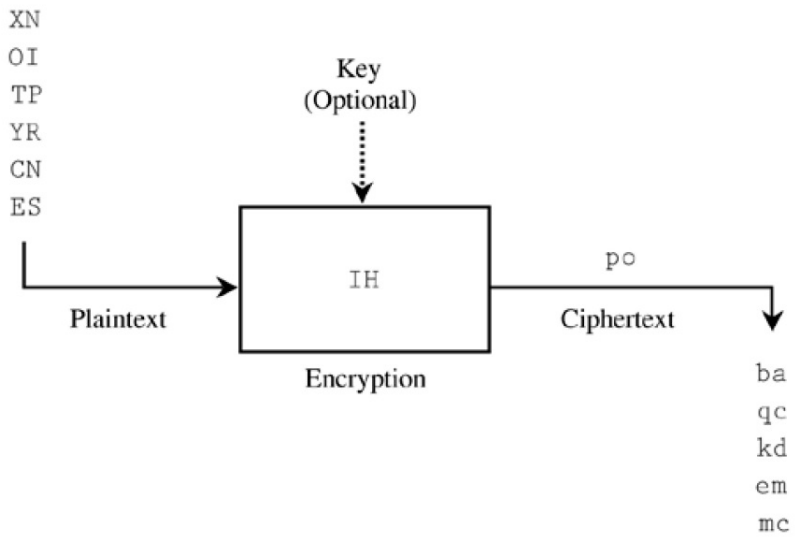

Figure 3.7: Block Encryption [49]

on block of plain text and convert it into a block of cipher text. Where stream ciphers are susceptible to malicious inserts, block ciphers are immune to it. Any insertion in the block will make the block length incorrect and the entire block will be deemed invalid.

3.1.5 Encryption Algorithms

This section will give a brief overview of what is an encryption algorithm along with a description of a couple of encryption algorithms. It will also discuss which encryption algorithms have been used in the project and reasons for their usage. As discussed earlier the process which transforms the plain text to an encrypted format is called as an encryption algorithm. For the sake of understanding, it can be viewed as some transpositions and jumbling which is reversible only when using a given key. There are several encryption algorithm available which have a varied level of security and key mechanism. There are several parameters which define the level of *security*

www.manaraa.com

or *trustworthiness* an encryption algorithm has. An algorithm which satisfies below qualities is deemed secure to be used [49] :

1. It is based on sound mathematics. The algorithm must have a mathematical proof of correctness and derived from solid mathematical principles.

2. Analyzed by experts and found to be correct. The algorithm which has been tested for issues by many experts is bound to be strong and resilient from any attack.

3. It has stood the test of time. An algorithm which has been around for a long time and still not broken, gains enough trust of users for its usage. A new algorithm, as it gains popularity, is used and tested by many people. It's limitations and mathematical foundations are reviewed time and time again. Although a long period of successful use is not a guarantee of a good algorithm, its flaws are generally discovered relatively soon after it's release [49].

Advanced Encryption Standard (AES)

Advanced Encryption Standard is a successor of another algorithm called DES (Data Encryption Standard) [21], a system developed by the USA government and officially adopted in 1976. It works on the principle of permutation and substitution and was an acceptable algorithm for general usage. Over the years as computing power of computers increased, it became easy to brute force DES algorithm, i.e. try every possible combination to break the code. AES overcame this susceptibility and was much stronger than its predecessor, DES. It was in 2001 that AES was formally adopted by US government [49].

AES is a symmetric encryption and work as a block cipher. Recall that symmetric encryption requires only one key for encryption and decryption. A major improvement

www.manaraa.com

in AES over DES is the key length. Key size is a critical parameter in determining the strength of any cipher. An encryption algorithm using a small key size can be *brute forced* easily.

Due to its popularity and given test of time AES has been the clear choice for encrypting the data in the research. Since the data is needed in blocks and a file may not always be in even block size, a padding has been used to be able to use AES for encryption. Details of the padding are given in implementation section of the manuscript.

Rivest-Shamir-Adelman Encryption (RSA)

RSA algorithm is an asymmetric encryption algorithm. Since it's introduction in 1978 it have been quite popular and remained secure. It based on the field in mathematics called number theory [33]. At the core of it lies the hard problem of factorizing large numbers. Since factoring large numbers is a problem for which no one has been able to find a shortcut or an easy method. It is considered as a solid basis of public key crypto system or asymmetric encryption.

There are two keys used in RSA d and e. These two can be used interchangeably but one must be kept private and another becomes public. A user which wants to share data with you must encrypt his message with your public key which only you can decrypt with your private key. E and D can be viewed as a complimentary function which can be applied in any order.

$$P = E(D(P)) = D(E(P))) \tag{3.1}$$

E and D can be thought as complimentary functions, which undo each other's effect. A plain text block is encrypted as $P^e mod n$. Factoring P^e to uncover the plain text is difficult since exponentiation is performed mod n. However, the intended

www.manaraa.com

receiver who knows the decrypting key d can easily retrieve P by computing $(P^e)^d$ mod n. The underlying principle of RSA is based on factoring large numbers. Until now, nobody has found a easy way to factor large numbers in a finite set, called a field [49]. It is considered to be a solid basis for a good crypto-system because this problem of factorizing large numbers has been around for many years and doesn't have an easy solution.

www.manaraa.com

Chapter 4

DATA PROTECTION ARCHITECTURE

The aim of data protection policy is to provide an end-to-end secure transfer of data and protecting the encrypted data from leaking or being accessed by unintended users/process. Traditionally, secure transfer of data is taken care by network protocols which can guarantee that data is not tampered over the network and is received by intended recipient. However, with increasing integration of cloud computing and cloud based apps, data is not secure even if it is present at the intended recipients desktop/machine. For protection over cloud and/or from service provider, we need to encrypt the data and create an infrastructure for the secure migration of the key to decrypt the data. I present below the components of data protection over cloud with a brief introduction as well.

4.1 Overview and Approach

In order to achieve the research objective as introduced earlier in the manuscript 1.4, a two pronged strategy is required. A part of the system which runs on the client machine which takes care of the security sub-system, and interacts with the Windows OS to keep track of the sensitive documents. This module requires to interact with the trusted server and authorized processes to expose the data. For the remainder of the section these modules would be generally referred as *client modules* as these are the software modules running on the client machine. The following sections will describe the intuition behind each module and how it relates to the objective.

The second section of data protection architecture comprises of services running at a trusted server which is responsible for the migration of the security key, autho-

www.manaraa.com

rization of users and users sharing documents. These features are required since our aim is to keep the security measures such that it does not hinders the usability of the system. A normal use case for a cloud enabled service is creating a document and accessing it at a different system without manually transferring it. Our proposed solution tries to not hinder this flow, but be able to intercept the data before the cloud process and allow a similar work-flow at the destination system. The server, referred as *migration authority* thereafter, makes sure that the document at the destination is readable only if the system/user is authorized. And the *client modules* at the destination do the remaining part of securing the data. To make the proposed solution more user friendly and usable, we also propose document sharing functionality, where users can share documents with each other, as a normal unsecured/plain-text sharing would do.

4.1.1 Protecting Data on the Disk

Protecting data on the disk can be considered as a first step toward building a granular security system. We first assessed which cryptographic algorithms to use to encrypt the data and then the ways to implement the encryption. In order to prevent data theft from any malware at the system, it is required to minimize any plain text version of our data. To achieve this we evaluated ways one can monitor the file system and protect the data on the disk by interacting with windows OS.

Monitoring File changes

In order to keep the data secure and protected from any attack it must be kept encrypted on the disk and only decrypted after authenticating the user and process requesting the data. In this section, I will describe how to encrypt the files whenever they are modified. This is required because the data should not be stored or exposed

www.manaraa.com

to cloud services in plain text. This relates with windows OS internals and file system services. This *File changes monitor* component keeps the track of the file on local hard disk and decrypts the data on the fly, after user authentication. There are more than one approach to let the data be encrypted on any changes. I will brief the approaches below and will give the details in the implementation section of the manuscript.

1. **ReadDirectoryChanges**

   The proof of concept utilized Read Directory Changes Windows API provided by Microsoft to monitor the disk activity of a particular folder. A very handy implementation by Jim Beverdge is available at his blog [45] which has been utilized in a prior version for the project. `ReadDirectoryChangesW` was introduced in Windows 2000 which reported what changes were made on the disk at a cost of added complexity. It was an advancement over previously available APIs like `FindFirstChangeNotification` which reported that something changed but did not report what changed. `ReadDirectoryChangesW` gets very complex as there are many combinations of I/O mode, waiting methods, and threading models [45].

   In our research, we used the above API to monitor a folder which contains the original documents, and whenever a change is made, the corresponding document in the cloud directory is updated, .i.e., the encrypted version of the document is uploaded on the cloud. Using this approach the we were able to prevent the plain text data be available to cloud and let it upload only the encrypted content. A major drawback of this approach is that it lets a copy of plain text data to be present at the system, hidden from the cloud but vulnerable to any malware on a compromised system. Another alternative which overcome

www.manaraa.com

this shortcoming is to use a file system filter driver as described in the next section.

2. **File System Filter Driver**

A quick and secure way of intercepting file system requests is by using a file system filter driver. It is not a device driver but an optional driver that lets one modify behavior of a file system or add value to it. A device driver is a software component that controls a particular hardware I/O device. E.g. a printer driver is used to communicate to the printer connected to the system. In contrast, a file system driver works in conjugation with file systems to control I/O operations. A file system filter driver can be used to *log*, *modify*, *delete* or even *deny access* to any request to the file system. It has typical application in anti-virus, encryption utilities and management system [41].

This proved to be an improvement over our previous approach for protecting data on the disk as it provided a run time control over the file changes. Next section will provide more details about file system drivers and where it aligns in the Windows OS stack.

User Space and Kernel Space At a broad level, a program execution in OS is executed in kernel mode or user mode. Memory in the OS is divided in two modes: user and kernel. Critical system process which require higher privileges e.g. *syscall* are executed in a secure mode called *kernel mode*. The user application or programs running in *user mode* do not have access to critical system instructions. This prevents user applications from accessing areas of the system which can lead to system wide instability. User application however can access kernel level function in a restrictive environment. Processes can switch between user mode and kernel mode by using system calls.

www.manaraa.com

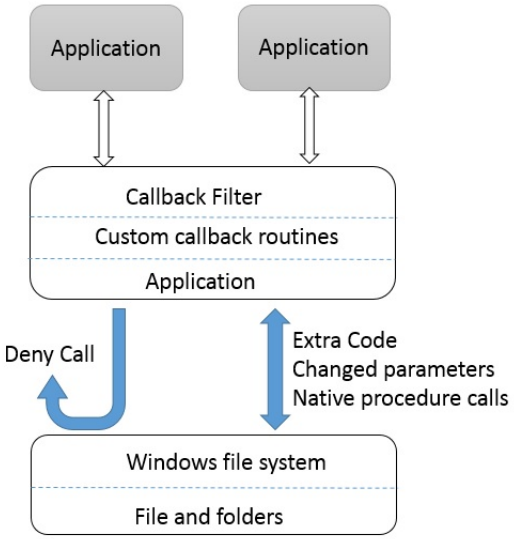

Figure 4.1: Eldos SDK [16]

Eldos Filter Driver [16] provides an SDK to build an user space driver to monitor and track disk activity. This has been used in the project to monitor the disk activity. Figure 4.1 provides an overview of the placement of Eldos filter driver in the windows OS stack and its interaction with other utilities.

Process Restriction Component

The above sections described how we are able to protect data on the disk but we also need to allow users/authorized processes to be able to use the data for reading and writing. This requires a component which would allow selective decryption and access to plain text data. The component *process restriction component* was created to achieve this goal. A brief functionality for this component is as follows. Users must be able to decrypt and access the data while the same data should not be allowed to be accessed by cloud service providers. This component restricts the cloud service provider process to get access to the decrypted data.

www.manaraa.com

For our proof of concept, `windows explorer` process [26] was allowed access to the data. Where as cloud service providers like `DropBox` [25], `GoogleDrive` [28], etc. where treated as unauthorized process. These processes would not be allowed to read decrypted data.

Encryption/decryption utility

Once we have control over the run time request for a file and established if we are going to go ahead or deny the request using the previous described module, we need to perform the actual task of encryption. This section describes the part of system which deals with cryptography and the ways it has been utilized for data protection.

In order to protect the data, so that only authorized parties can have access to it, encryption is used to convert the data in a form which only the authorized parties can convert back to a readable form. Converting the data to a meaningless format is called encryption and converting it back is called decryption. A *key* is a construct which is the link between the two formats.

There are various ways one can protect the data on the disk. One of the popular methods is to encrypt the data on the entire disk. This method does provide the needed protection but it takes a toll on the system and treats the entire data in a same way. Most businesses protect their laptops using this scheme but it puts the entire security responsibility in the enterprise's hand and does not give the flexibility and much needed selectiveness in securing the data as discussed in section 2.3.1.

To keep the process quicker and not impact the performance we use a encryption scheme, `digital envelope` [23]. It is a secure electronic container which is used to protect data through encryption. It utilizes the speed of symmetric encryption algorithm and security of asymmetric encryption algorithm. A description for these

www.manaraa.com

algorithms is already provided in chapter 3, here we discuss its usefulness for our requirements.

The major drawback of asymmetric ciphers are the speed and security strength [60]. An asymmetric cipher is based on computational problems which determine its key strength, e.g : The RSA algorithm is based on Integer Factorization Problem(IFP) [30], Diffie-Hellman algorithm is based on Discrete Logarithm Problem (DLP) [24]. The computational cost makes asymmetric algorithm less efficient and takes up much resources [51]. Symmetric algorithm on the other hand is faster but needs the secret key to be secure.

In order to maintain the security and performance, a hybrid mechanism is needed which over comes the problem we face with the above mentioned algorithms. The bulk of encryption is done using a symmetric key, which is fast, and the symmetric key itself is encrypted using public key cryptography and stored in the file header. The public key cryptography utilizes Intel's IPT with PKI infrastructure to securely decrypt the symmetric key. This approach gives the required performance benefit and maintains the needed security.

4.1.2 Enabling Mobility

The previous section described how we are able to protect data on the disk. In this section I will detail about enabling the required mobility of the protected data. The encrypted document is uploaded on the cloud and it needs to be decrypted at the destination. Due to the *client modules* present at the system, encrypted content is uploaded on the cloud which takes care of the transfer of the data. A key migration system is designed which would allow a secure decryption of the encrypted data. The below section describes the way it has been implemented satisfying our requirement.

www.manaraa.com

Key Migration System (KMS)

In order to securely decrypt the data, the encryption key is needed at the destination. This component is responsible for migrating the user key from source to destination. It is designed to be an enterprise on-premise server that interface with the Enterprise Identity Protection. The server must be a trusted machine and highly secure. Securing a single machine is easier than securing the individual clients connecting to the server. We have designed the service to be similar to a server-client architecture which could be run as an IIS service on an enterprise network. It needs to support both push and pull model since a new document can be created as well as an existing one need to be decrypted on a new machine. The following section will describe the responsibilities the service has been designed with, along with the motivation for each functionality.

1. **Client Authentication**

   A major requirement for connection to KMS is authentication. Server need to authenticate the client in order to start the communication. It needs to be more secure and this is where Intel IPT-PKI plays an important part. It provides a firmware enabled authentication making sure that client is connecting from an authorized system and a malware is not trying to spoof the system. This technique is part of trusted computing technology which enables hardware based protection loading it with a special encryption key inaccessible to rest of the system [39]. This is achieved using certificates, a process called *attestation*, and encrypting a *nonce* which gives the proof that the server has the correct key. The authentication process is described in detail in section 4.3.

2. **Client Attestation**

www.manaraa.com

The server needs to verify if the client has the correct key and the client is in fact a trusted machine in order to start the import/export of key. This is done with the help of Intel IPT-PKI which gives the proof of key being generated in the firmware. This module takes care of interacting with firmware to make sure the system is authentic. *Intel IPT Attestation* [59] aims to provide evidence that the key is protected by Intel security engine. It uses enhanced privacy identifier(EPID) to achieve this. EPID is an asymmetric key embedded into Intel hardware at manufacturing time.

3. **Key storage at server**

   The keys must be stored at a trusted machine acting as server. Server stores the keys and the user mapping. It also defines the operation type which the client will perform. If it does not have user key the client needs to generate a user key and export to the server. Depending on the presence of key, the client perform import or export operation.

4. **Database Management**

   The centralized highly secured server is required to manage various Keys and identification for the users. As the number of users grow a database is needed to store the following information important for migration and maintenance of the users. For our proof of concept we used a serialized hash map to store the information. The server needs to store some information in order to uniquely identify the resource and the user it belongs to. This information is looked up when a request for a document key is made later in the process. Below are the fields which were stored at the server:

   User or GROUP ID : This is a unique identifier to identify a user. This identification can also be used to enable group level access or a single user access.

www.manaraa.com

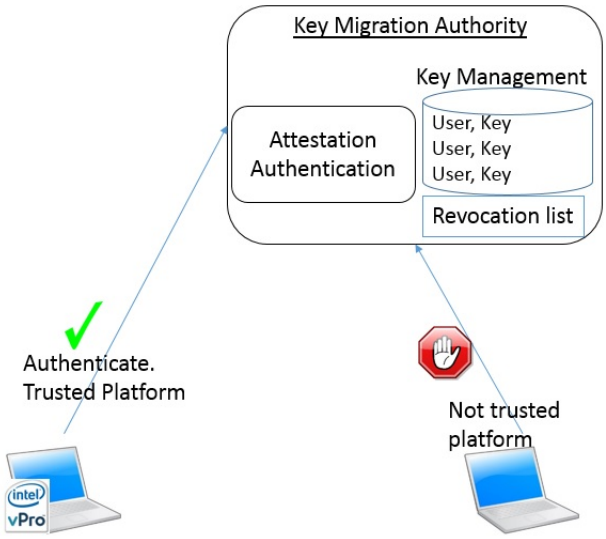

Figure 4.2: KMS Responsibility

GUID or resource ID: This is a unique identifier for a resource. A resource could be a file or a folder located at a client destination, it serves as a name for a protected entity. The GUID is also used as a key container name while communicating with crypto service providers at client.

5. **Maintain revocation list**

   The server also needs to maintain the list of users who have been revoked. In case of user revocation, his account and corresponding keys must be invalidated. This prevents users from accessing the document when a user has left the organization.

Figure 4.2 depicts the above discussed responsibilities pictorially. E.g. KMS will reject a connection request from system which is not running on a trusted platform. It also stores list of users and their key for authentication or rejection if the key is expired or invalid.

www.manaraa.com

4.2 File Header

File header plays an important role in enabling the security system. Each file should have a separate key which needs to be attached with it along with other required options. This data is stored in the first block of the file as a file header. Meta data about the file is stored in the file header which allows to identify the file and extract the session key for the file. The file header also helps in adding any information which will be helpful in adding more functionality. E.g. information about the parent folder on the system is also stored in the file header which is used to group files in a particular folder and provide folder level flexibility.

The file is encrypted using a symmetric key which itself is encrypted by the user key and stored in the file header. Figure 4.3 shows the structure of the file pictorially. In another version of the solution File Header has been modified to include a specific GUID so as to identify the files for particular folder.

Fields stored in file header

1. **Container:**

   This is a unique identifier which identifies a file or folder. This is also used as a container for storing the key using MS cryptography APIs.

2. **Key:**

   The key used to encrypt and decrypt the content

3. **Initialization vector (IV):**

   IV is an arbitrary number often called *nonce* [19], which is used along with the security key for encryption [31]. It's makes it difficult for any hacker to utilize dictionary attacks [22] to crack the encrypted data.

Structure of File Header Class:

www.manaraa.com

Figure 4.3: File Header Overview

1. **Key**: This field stores the key used for encryption and decryption.

2. **IV field**: This field stores the IV associated with the key.

3. **Parent Folder**: This field is used to keep track of the folder in which the file reside. This is helpful in identifying and grouping all the files under one folder identifier.

4.3 Key Export Procedure

The export mechanism deals with securely connecting to server and storing the key for its usage for decryption at the destination. The client module connects to migration authority module and authenticates itself before exporting the key. It follows below steps for key export:

1. The client acquires it's public key and sends it to the server along with user's credentials. The server verifies user's credentials and public key. This is a two-step verification. First the user's credential, *what the user know* and second is the authentication of trusted system *trusted computing*. The user's credential can be the cloud service provider credentials or an active directory credential.

www.manaraa.com

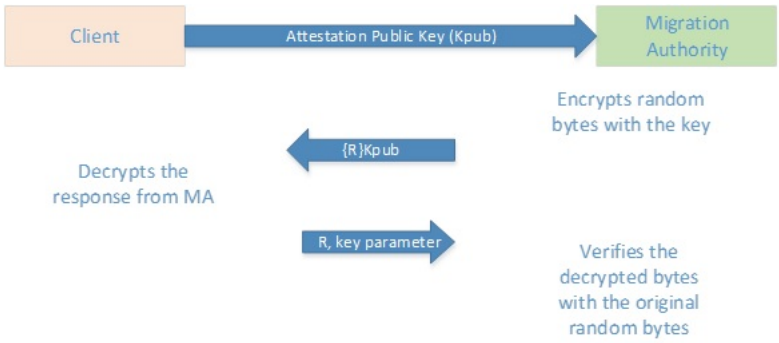

Figure 4.4: Key Exchange

To verify for cloud credential, OAuth v2.0 protocol [34] is used. For enterprise using active directory [11], authentication is done using LDAP protocol [12].

2. The server then encrypts a random set of bytes *nonce* with the received public key from the client and sends it back. Server also creates a user session and stores the user info along with the public key. The public key of user is converted to PKCS 11 format (public key cryptography standards) [35] used to communicate to hardware crypto system.

3. The client on receiving the encrypted data, decrypts it using it's corresponding private key and sends the nonce back to the server for verification. Server verifies the key and determines if client can perform export operation. Server then sends a certificate to the client for exporting the key.

4. Client reads the certificate and extract the server's public key to export its key in a PKCS7 blob [36].

5. Server reads the RSA key in the exported blob and stores in a file/database for the user. Over the course of time, as the number of keys increases, it can be stored in a database instead of file. For our research, we chose to use files to store information.

www.manaraa.com

4.4 Key Import Procedure

Similar to Key export procedure key import is also done after first step verification of user's credentials and client public key. After successful completion of the handshake protocol as depicted in 4.4 and described in section 4.3, secure import is processed. The client sends the key container name and its certificate for import and server sends the key and it is imported in the Intel CSP container.

4.5 Key Sharing

The encrypted file should be able to be decrypted at the client upon successful authentication of the user. This is a two step process, first the user must provide his credentials, cloud or Active directory. This is to let him access his files or files to which he is authorized. Second is the Firmware based key authentication using Intel IPT with PKI which lets the user decrypt the encrypted document. This is taken care by the Key Migration Authority. This model allows a user to upload his data over cloud and use it on other system. However, in a normal day to day scenario, documents are shared by people. This creates a need to share the corresponding secret key to the authorized users.

The key sharing modules lets the users share the document with another user in their organization and takes care of providing the key to this user. Figure 4.5 shows a pictorial representation of key sharing mechanism. The sharing process is carried out as below:

1. User1 shares a folder or file by specifying another user's email.

2. Migration authority updates this information at a secure and central repository.

www.manaraa.com

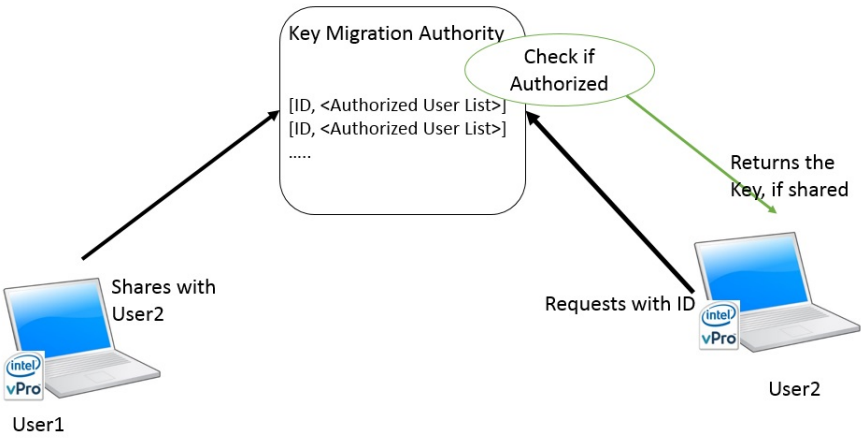

Figure 4.5: Key Sharing Overview

3. User2 connects to KMS using the same two step authentication process as described earlier. User1's encrypted file is transferred to User2 over cloud. KMS is not responsible for transferring the entire file itself. It is only responsible to provide the key to decrypt that file upon successful authentication for user2. KMS upon user2 request for decryption checks if the user has been authorized to this file. User2 email should be same as specified by User1 for the requested file in the previous step.

4. If authorized, the KMS exports the key and the client service at user2 securely decrypts the file.

www.manaraa.com

Chapter 5

IMPLEMENTATION AND EVALUATION

The proof of concept was implemented using .Net platform (C#) [4] and uses Microsoft Cryptographic APIs for encryption. It also uses Intel CSPs (Cryptographic Service Provider) [20] and DAL (Dynamic Application Loader) [7] for interacting with underlying firmware and cryptography services.

5.1 Protecting Data on Disk

A major problem with data protection is avoiding plain text data present on the system to be accessed by cloud service provider. The implementation using `Read Directory Windows API` used an approach where the cloud directory is synced with a protected directory which has a corresponding unprotected directory. User make changes in the unprotected directory and the corresponding file in protected directory is updated with the changes - encrypted. This way, plain text is not exposed to cloud but an encrypted copy of it is uploaded on the cloud. Any subsequent updates for the documents is encrypted and synced with the cloud directory. This approach has a disadvantage that it lets the plain text get stored in the file system where it is susceptible to any virus or malware attack.

A better approach is to use a file system driver to encrypt and decrypt the data whenever a file is saved or updated. This way data stored on the disk is always encrypted and decrypted only when there is a legible request for read. This approach gives more control over windows OS file system requests. We have used Eldos file system filter driver [16] to control the request to file system and hook up our encryption and decryption system.

www.manaraa.com

Load order group	Level	Description
FSFilter Encryption	140000-149999	This group includes filter drivers that encrypt and decrypt data during file I/O.

Table 5.1: Load Level for FS Filter Driver [32]

5.1.1 Consideration for Other File System Drivers

Windows mini filter architecture allows multiple file system drivers to run each with an altitude of its own. E.g. other file system driver present on a system, anti-virus or a disk encryption utility, can interfere with our file system driver if not configured correctly.

During our research, we faced similar problem when our file system driver was giving incorrect results due to interaction with anti-virus present on the system. The resolution was to use Eldos file system driver in a mini-filter mode with a specified level according to Microsoft definition.

Microsoft Windows XP and later OS provide a dedicated set of load order for file system driver that are loaded at system start up time. A filter can only be attached on the top of the stack of the already attached filter and can not be inserted in the stack. Hence the order of filter load is important [32].

5.1.2 Virtual File

One problem with using file filter for runtime encryption is when we need to selectively send the data for the file. That is, for one process we want the encrypted data and another process to have the data decrypted. A simple approach that comes

www.manaraa.com

to mind to achieve this is to check for the calling process, and perform decryption or not decrypt and the request go through. However, the problem with this approach is related to the design of OS cache. Once a file is accessed the data is stored in the cache and provided to the next application which requests for it. Currently we do not have control over this behavior of OS cache and made use of virtual files to allow for selective decryption.

Virtual file comes in handy to give a work around for this problem. For one file A.txt a second dummy file is created, A.txt.enc, which simply acts as a placeholder to redirect calls to the restricted process. Let's say one does not wants `Dropbox` to access decrypted content. Call back filter can achieve this, it can restrict showing selected files to any processes. Using this, `Dropbox` views A.txt.enc as A.txt and sends the request to access it. However, the request sent to OS is for A.txt.enc which is different than A.txt, Hence the problem with cache is circumvented and the request for the file can be intercepted separately and the data passed as encrypted. While for the authorized processes, the normal file A.txt will be access and decrypted.

5.1.3 Sample Flow: File Creation

Figure 5.1 shows how a User action triggers the system which intercepts the calls and subsequently invokes underlying layers for further actions. The figure shows a sample call routine when a User creates a file in the protected directory. Filter driver intercepts the call to file system and provides a Call Back function (write call back) which is then queried with Intel IPT-PKI CSP for user's key. If the user key is not present an import procedure is triggered which imports the encrypted key from the key migration authority and stores in the CSP. Using this CSP, a new symmetric key is generated to encrypt the file and encrypted with user key and stored in the file header for future encryption and decryption.

www.manaraa.com

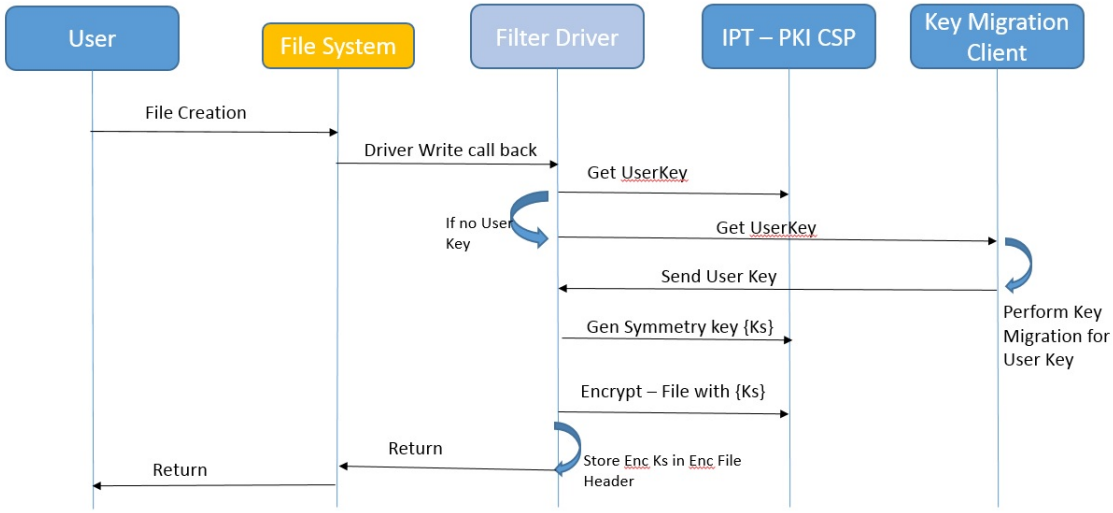

Figure 5.1: File Creation Flow diagram

5.2 Data Protection Service

It is the core module which is responsible for monitoring the file system and connecting to the server for secure key migration. It is designed to run as a service at the client which runs in background with very less user interaction. The reason it has minimal user interaction is because security application should not be much intrusive such that it affects the productivity of users. The service upon start, instantiate and registers the file system driver and sets the callback to intercepts.

5.2.1 Encrypt Context Class

Each file in the protected directory is associated with an encrypt context to store the meta data and key information. This is represented by encrypt context class. It stores the following information:

1. Session key and initialization vector.

www.manaraa.com

2. File information using WIN32 structure.

3. File Header having the encrypted keys and GUIDs.

4. Buffer Size:

   Size of the buffer which is read from the disk. This is used in reading the header from the file stored on disk and properly offsetting further reads.

5. Reference Counter:

   Used to free the memory if it is not in use by any object anymore.

5.2.2 File Header Module

This module is used to read and store the file header information from the file stored on the disk. It stores key and the IV. The structure also has a flag variable which helps decide if the file already has a header or a new header is required to be generated for a new file. Whenever a file is opened or edited, file header class is instantiated and the three fields are stored: container, key and IV. It is also responsible for starting a session with the migration authority in case the client does not have the key to decrypt the user key for the file (the case when a user receives a shared file in his cloud directory). It requests the migration authority for the key, providing the user information and attestation public key with a nonce, following the authentication process as described in Chapter 4, Section: 4.4.

5.2.3 File Filter

This section explains how Eldos file filter is implemented in the project. The API provides callbacks functions which are used to call our specific functions. API is invoked by calling `SetRegistrationKey()` and `InstallDriver()`. Eldos provides two classes of callbacks:

www.manaraa.com

1. Notify CallBack :

   These callback are used to notify of any operation that has been performed. These have their application in audit application which only keep track of the activities.

2. Control CallBack:

   These callback lets the application take control of the execution of some operation. These function intercept the request before it is passed to the file system. These are used in our project to deny or modify any file system request.

The following section gives a short description of few important callbacks used in the project below:

1. `CbFltReadFileC`

   This callback function is called when there is request for a read of a file. The function has buffer location, size to read and the start location. At this point the data from file system has not yet been read but it is used to make sure the position and the number of bytes to read are offset by correct position. Offset is required because we do not want to pass the header of the file to the application requesting the data for the file.

2. `CbFltPostReadFileC`

   The callback provides the number of bytes which are to be read and the location of the buffer. Using this information the buffer is decrypted and the modified data is stored in buffer and the request is passed to the OS.

3. `CbFltWriteFileC`

   This callback is called after a file is written and data is modified. Similar to read callback it provides the buffer, location and bytes to write. The data in

www.manaraa.com

the buffer is overwritten with the encrypted content and request is passed to the underlying file system. Care is taken while encrypting data such that the buffer size is not offset by the encrypted data. To take care of this problem, two approaches are used:

(a) Use stream cipher

Stream cipher converts one plain text character to encode character. This gives te encrypted data size which is same as the input data size. However stream cipher is not very secure and prone to error as described in previous Chapter 3 on cryptography.

(b) Use block encryption

To increase the strength of encryption, block encryption is used to encrypt buffer data. However, since buffer size is fixed and data is accessed in multiples of buffer size. A padding is used to make the data size equal to that of required block size by the encryption algorithm

(c) Use Block Encryption in counter mode.

A better approach is to use a block encryption algorithm, AES in counter (CTR) mode. Counter mode turns a block cipher in to stream cipher by using a counter which is used to generate the next key stream. The counters are generated such that they are not derivable from last values for long, if at all [64].

Using AES encryption algorithm in CTR mode gave us flexibility with the data size and also the existing security of block encryption algorithm. This scheme did not require us to add a padding to the last chunk of data in order to use AES block encryption. CTR mode also allows to access and decrypt random bits which gives more flexibility in data decryption.

www.manaraa.com

5.3 Migration Authority

Migration Authority is hosted as a service on a trusted computer and built as a WCF service. Important function it exposes to the client are:

1. **Login**

   This function is the initiating function and has to be called first when connecting to the server. It takes the user credentials and the public key and returns a nonce encrypted with the user's public key. It also stores the public key, user and the type of operation to be performed in a user session at the server. Type of operation is set to unknown at the first login, but changed to import or export depending on the presence of the key at the client.

2. **Export**

   This function extracts the input key envelope and stores the key at the server for the container name passed as input.

3. **Import**

   This function wraps the key for the user in a pkcs7 envelope [36] using the received certificate and the user/container name.

4. **Authorize User**

   This function is responsible for sharing the file between users and determines if a user requesting for a file has been given permission to access. This information is specified by the user sharing the file.

5. **VerifyKey**

   This function decides which operation to perform at the client side. The server decides whether to do a key push or pull by checking if it has a key present for

www.manaraa.com

the connected client. If the server has the key for client, a export operation is performed, else an import operation is done. This avoids the explicit request from client for key import or export and limits the affect of rogue clients.

5.4 Evaluation

The aim of data protection is to provide an end-to-end security system which can be used by the users without being too intrusive and not impacting the performance adversely. We evaluated the system such that it satisfies our primary motive of protecting data from cloud service provider and tested the migration of security key. We evaluated the system on varying size and type of files and time taken to import or export the key to ascertain the feasibility and practicality of the approach.

Our solution provides a novel way to protect your document which is not exposed in clear to cloud service provider. The solution also makes it possible to take the control of the key for the protected document. The key is firmly kept protected in the firmware and is only exposed at the destination system with the same secured system. This achieves the objectives we described in section 1.4

1. Decryption only on selected platform

   We tested the solution such that it decrypts only on the systems enabled by Intel IPT-PKI technology. An experiment was set up to create and upload data on the cloud using our system where both the end system where enabled with Intel vPro technology. We verified the following:

   - The documents uploaded on the cloud were encrypted.
   - The documents on the end system were only accessible while our encryption system was running.
   - The document were decrypted correctly at the other system.

www.manaraa.com

We also verified that our system did not decrypt the data on non Intel IPT-PKI enabled systems thus increasing the security of the data.

2. Encryption Verification

   The solution was verified on various inputs to verify it's security. We have used AES in counter mode which allowed us to test with random size of data. Although reverifying a cryptographic algorithm is not feasible but we tested our solution to make sure we have the right implementation. Below types of data set was input to the system and the output cipher text was verified to not contain any patterns as such.

   - Random input data
   - Sample test data
   - Stream of 0s. and same data

   The table 5.2 displays few sample inputs to show the encrypted content. Please note that not all characters of cipher text are in printable format so the corresponding hex values are also given.

3. Encryption performance

   We evaluated the performance of our encryption system on multiple iteration of test data. The performance was validated against the below formula 3.

$$T = k * n + i + f \tag{5.1}$$

   where:

   (a) T is the encryption time

www.manaraa.com

Plain Text	PlainText Hex	Cipher Text	CipherText Hex
Encrypting same data			
test	74 65 73 74	%6G	25 C2 36 47
test	74 65 73 74	p ɹΣ	70 A5 11 06
test	74 65 73 74	ffi ° -	0E EE 17 20
Sample word test.	53 61 6D 70 6C 65 20 77 6F 72 64 20 74 65 73 74 2E	a.UP.rh	61 1B 55 50 88 12 AE B2 72 C3 E4 68 CB FD 92 BC A5
Encrypting Random Bytes			
@.L-5%H.	40 00 4C D0 2D DB F3 35 25 48 F1 1D 20 89 A4 CA	.”c..$O4‘	90 22 63 D9 02 8C BE 15 24 4F 34 BA 9C 83 C7 60

Table 5.2: Sample Test Results for encryption

(b) n is the input data size, in bytes

(c) i is the processing time needed to begin the processing.

(d) f is the time needed to wrap up the encryption. i.e. after the last encrypted bit it output.

(e) k is the asymptotic encryption speed.

To evaluate the encryption system, various size of data was encrypted repeatedly to note the timings. Since the performance depends heavily on CPU usage and hardware. System was evaluated at various instances to asses generic performance. The configuration for the test system was as below:

- **Processor:** Intel(R) Core(TM) i5-5300U CPU @ 2.30GHz 2.30GHz
- **Memory:** 8.00 GB
- **System type:** 64-bit Operating System, x64-bit processor

Time was measured using *StopWatch* Class provided by .Net platform. *StopWatch.ElapsedTicks* property gives the elapsed time measures in timer ticks. It is

www.manaraa.com

i	f	$k * n$	**ticks**	**Time(ms)**	**size (KB)**
1668.3	27.2	4088.4	5783.9	2.58105081464709	10
1746.5	41.2	4248.5	6036.2	1.89588242985324	256
1674.8	46.3	3986.8	5707.9	1.77909946365515	512
1831	74.8	4192.8	6098.6	1.87102644507207	1024
1746.8	112.7	4106.3	5965.8	1.83242603782661	2046
2274.7	247.2	5797.9	8319.8	2.58729827940358	4092
2119.6	383.7	4921.9	7425.2	2.19638548464038	8184
4341.1	2503.8	8821.3	15666.2	3.93648291831574	16368
3627.1	1369.8	5396.7	10393.6	2.4082637893819	26188
6580.7	2264.9	8191.8	17037.4	3.65557012801502	32736
7161.5	3643.7	6447.8	17253	2.87731451834947	65472
12773.1	6717.8	6738.5	26229.4	3.00703866154315	130944
32688.1	25478.3	13538.4	71704.8	6.04147691851834	523776
61798.7	50205.5	20464.6	132468.8	9.1322762325467	1047552
176552.9	122156.9	68443.3	367153.1	30.5426503262738	2095104
280575.2	218752	99805.7	599132.9	44.5380423747685	4190208
510962.5	397516.1	129538.7	1038017.3	57.8063187751042	8380416
984262.4	804122.1	285990.6	2074375.1	127.622585299091	16760832

Table 5.3: Encryption time over various byte size

system hardware dependent and is the smallest unit of time that StopWatch can measure. It is converted to absolute time by using *StopWatch.Frequency* field which indicates timer precision and resolution. Frequency of the test system was 2240909 ticks per second, which resolves to (1/2240909) seconds per tick.

The results show the amount of time spent in various stages, i.e. initializing the data and crypto algorithm, performing actual encryption and wrapping up the encryption. Figure 5.2 shows the break up of time taken in various stages in the encryption. One can analyze that most of the time is taken in initializing and finalizing

www.manaraa.com

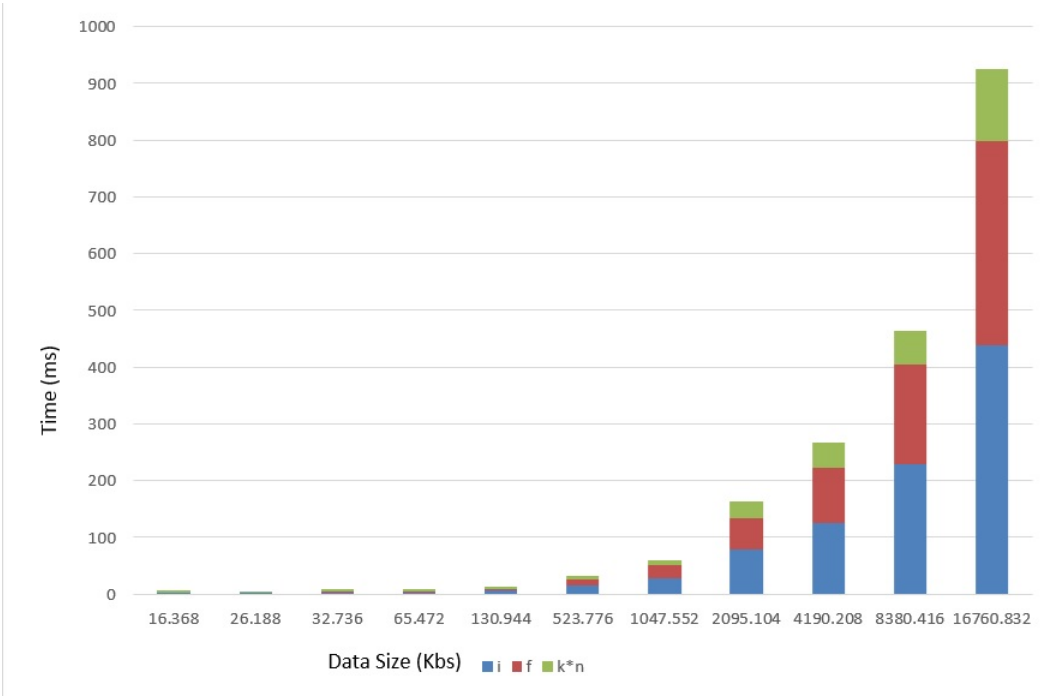

Figure 5.2: Encryption Time Analysis

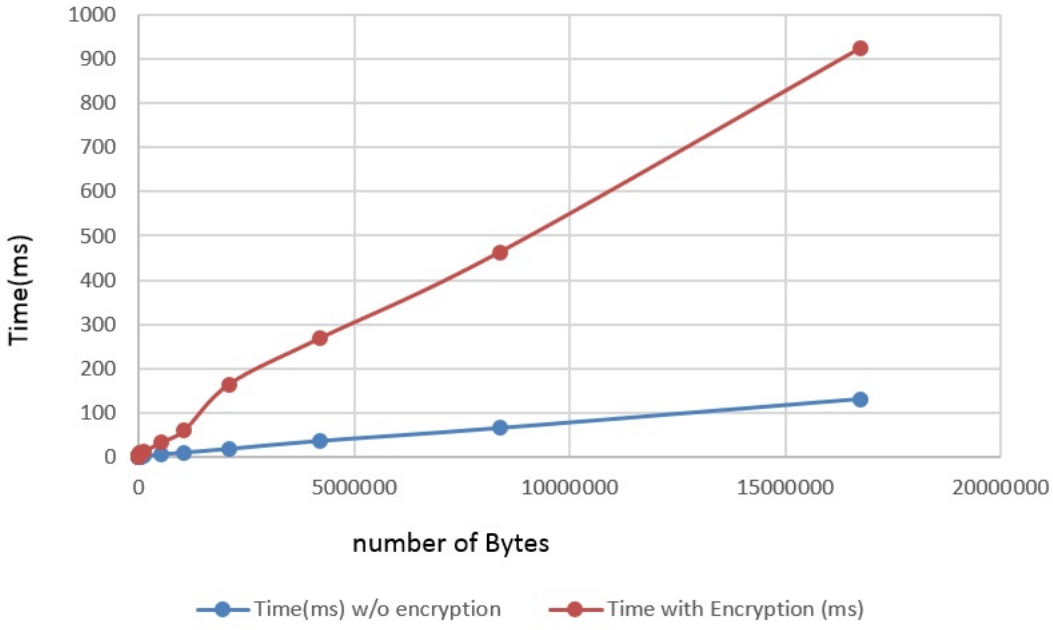

Figure 5.3: Time With and Without Encryption

www.manaraa.com

stages, which explains the crypto algorithm which is been used in the implementation. We have used AES algorithm in CTR mode where we first pre-process the plain text block using a nonce and a block number. The cryptographic object is initialized and then the actual encryption is performed. After encrypting the processed buffer, the resultant buffer is simply XORed with the input and the encrypted block. This results in the output buffer, whose size is equal to that of input buffer. Figure 5.3 shows a comparison between the time taken to encrypt data buffer versus time taken just to process the buffer without encryption. Encryption does take a hit on the performance but it is still in order of milliseconds and grows as the data size increase along with the time without any encryption.

5.4.1 Key Import and Export Timings

In a experimental set up, timings were noted for the import and export using the proof of concept. The machines were Intel Vpro platform based and supported Intel IPT-PKI infrastructure. The timings were measured on a typical enterprise laptop with SSD HardDisk and 8 GB RAM. Table 5.4 show the timings. These timings were noted while running the system on debug mode so they also involve the debug and logging time.

www.manaraa.com

Function	Time (ms)
GetPublicKey	980
Login	98
DecryptWithKey	365
Get Attestation	2137
VerifyKey	10
GetCertiifacteforImport	535
Get	9090
Export	110
Total	13421

Function	Time (ms)
GetPublicKey	317
Login	5126
DecryptWithKey	471
Get Attestation	214
VerifyKey	4
GetCertForImport	519
Import	110
SetPKCS7Envelope	796
Total	9594

Table 5.4: Time Taken for Export and Import Operation

www.manaraa.com

Chapter 6

CONCLUSION AND FUTURE WORK

With data attacks getting common these days the proposed security architecture which utilizes hardware based protection makes a strong defense from the recent attacks. The research implemented a proof of concept for securing the data and making it attack resistant using Intel based IPT-PKI infrastructure. This feature gives an improvement over other similar solutions which do not have a hardware based security. The manuscript talked about recent attacks and how threats to information due to increased usage of cloud rises. A novel solution using hardened security was presented and architecture was discussed. This report also described applicable theoretical concepts and usage of cryptography in protecting the data. The research also implemented file system filter driver to monitor data on hard disk and provide run time protection from cloud service provider.

The solution has many use cases which make this concept useful to various scenarios. It has its usage in web email based encryption. Web based email (yahoo [42], gmail [27] can be encrypted using enterprise based criteria which gives another level of protection.

Another application is in health care industry where a health care personnel has to interact with many systems and enter his/credential across various system. Using the presented concept, the login information can be securely transferred to multiple devices and is tamper proof on unauthorized system.

www.manaraa.com

6.1 Further Work

The proposed solution in our research protects the data at rest and in transit. However while the data is in memory, it is still susceptible to a malware running on escalated privileges. The data need not be protected only when it is static but also during run time. To increase the level of protection another technology, Intel SGX can be used which protects the sensitive data from unauthorized access or modification by rogue software running at higher privilege levels [54].

Data Security is a field where new attacks would be discovered with passage of time and new methods of prevention will need to be invented. And as such, no software can guarantee 100% protection from any attacks. But with innovation and research, attacks can be mitigated and information can be secured.

www.manaraa.com

[1] "Gartner predicts by 2017, half of employers will require employees to supply their own device for work purposes", URL http://www.gartner.com/newsroom/id/2466615 (2013).

[2] "How secure is microsoft skydrive?", URL "http://www.tomsguide.com/us/how-secure-microsoft-skydrive,review-1815.html" (2013).

[3] "State of byod and mobile security report latest insights trends and stats", URL "https://securityintelligence.com/state-of-byod-and-mobile-security-report-latest-insights-trends-and-stats/" (2014).

[4] "C #", URL https://msdn.microsoft.com/en-us/library/kx37x362.aspx (2015).

[5] "Cmu graphics lab motion capture database", URL http://mocap.cs.cmu.edu/ (2015).

[6] "Examining the security of cloud-based vs. on-premise deployments", URL http://www.infor.com/content/industry-insights/security-of-cloud-vs-on-permise-deployments.pdf/ (2015).

[7] "Intel identity protection technology with pki", URL http://www.intel.com/content/www/us/en/architecture-and-technology/identity-protection/identity-protection-technology-general.html (2015).

[8] "Target hacked", URL http://www.huffingtonpost.com/news/target-hacked/ (2015).

[9] "Usa patriot act effect on cloud computing services", URL "http://www.itlawgroup.com/resources/articles/113-usa-patriot-act-effect-on-cloud-computing-services" (2015).

[10] URL http://windows.microsoft.com/en-us/onedrive/skydrive-to-onedrive (2016).

[11] "Active directory", URL https://en.wikipedia.org/wiki/Active_Directory (2016).

[12] "Authentication using ldap", URL http://www.tldp.org/HOWTO/LDAP-HOWTO/authentication.html (2016).

[13] "Better cloud", URL https://www.bettercloud.com/ (2016).

[14] "Boxcryptor", URL https://www.boxcryptor.com/en (2016).

[15] "Bring your own device", URL https://en.wikipedia.org/wiki/Bring_your_own_device (2016).

www.manaraa.com

[16] "Callbackfilter", URL https://www.eldos.com/cbflt/ (2016).

[17] "Cipher cloud, trust in cloud", URL https://www.ciphercloud.com/ (2016).

[18] "cryptanalysis", URL http://searchsecurity.techtarget.com/definition/cryptanalysis (2016).

[19] "Cryptographic nonce", URL https://en.wikipedia.org/wiki/Cryptographic_nonce (2016).

[20] "Cryptographic service provider", URL https://en.wikipedia.org/wiki/Cryptographic_Service_Provider (2016).

[21] "Data encryption algorithm", URL https://en.wikipedia.org/wiki/Data_Encryption_Standard (2016).

[22] "Dictionary attack", URL https://en.wikipedia.org/wiki/Dictionary_attack (2016).

[23] "Digital envelope", URL https://www.techopedia.com/definition/18859/digital-envelope (2016).

[24] "Discrete logarithmic problem", URL https://en.wikipedia.org/wiki/Discrete_logarithm (2016).

[25] "Dropbox - file storage and backup", URL https://www.dropbox.com/ (2016).

[26] "Explorer", URL http://www.paretologic.com/resources/paretolabs/exe/explorer_exe.aspx (2016).

[27] "Gmail", URL http://searchsoa.techtarget.com/definition/Gmail (2016).

[28] "Google drive - a safe place for all your files", URL https://www.google.com/drive/ (2016).

[29] "A history of cloud computing", URL http://www.computerweekly.com/feature/A-history-of-cloud-computing (2016).

[30] "Integer factorization", URL https://en.wikipedia.org/wiki/Integer_factorization (2016).

[31] "Intialization vector", URL https://en.wikipedia.org/wiki/Initialization_vector (2016).

[32] "Load order groups for file system filter drivers", URL https://msdn.microsoft.com/en-us/library/windows/hardware/ff549694(v=vs.85).aspx (2016).

[33] "Number theory", URL https://en.wikipedia.org/wiki/Number_theory (2016).

www.manaraa.com

[34] "Oauth 2.0", URL http://oauth.net/2/ (2016).

[35] "Pkcs 11", URL https://en.wikipedia.org/wiki/PKCS_11 (2016).

[36] "Pkcs 7: Cryptographic message syntax", URL https://tools.ietf.org/html/rfc2315 (2016).

[37] "Rsa (cryptosystem)", URL https://en.wikipedia.org/wiki/RSA_(cryptosystem) (2016).

[38] "Sony pictures entertainment", URL http://www.sonypictures.com/ (2016).

[39] "Trusted computing", URL https://en.wikipedia.org/wiki/Trusted_Computing (2016).

[40] "What is a file system filter driver?", URL https://msdn.microsoft.com/en-us/library/windows/hardware/ff557282(v=vs.85).aspx (2016).

[41] "What is a file system filter driver?", URL https://msdn.microsoft.com/en-us/library/windows/hardware/ff557282(v=vs.85).aspx (2016).

[42] "Yahoo mail", URL https://overview.mail.yahoo.com/ (2016).

[43] Abrams, M., *A Perspective: Data Flow Governance in Asia* (2008).

[44] Benameur, S. P., *Privacy, Security and Trust Issues Arising from Cloud Computing.* (Dec. 2010).

[45] Beveridge, J., "Understanding readdirectorychangesw - part 1", URL http://qualapps.blogspot.com/2010/05/understanding-readdirectorychangesw.html (2010).

[46] Brinkman, R., *Searching in encrypted data*, Ph.D. thesis, URL http://doc.utwente.nl/57852/ (2007).

[47] Carbin, P., "Intel identity protection technology with public key infrastructure", URL https://software.intel.com/en-us/articles/intel-identity-protection-technology-based-token-provider#key (2015).

[48] Carey, P., "Intel beats the street again", URL http://www.mercurynews.com/business/ci_29385915/intel-beats-street-again (2016).

[49] Charles P. Pfleeger, S. L. P., *Security in Computing, Third Edition, ISBN: 0-13-035548-8* (2003, 1997, 1989 Pearson Education, Inc.., 2015).

[50] Chow, R., P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka and J. Molina, "Controlling data in the cloud: outsourcing computation without outsourcing control", in "Proceedings of the 2009 ACM workshop on Cloud computing security", pp. 85–90 (ACM, 2009).

[51] Dr. Prerna Mahajan, A. S., "A study of encryption algorithms aes, des and rsa for security", (2016).

www.manaraa.com

[52] Esq., K. N. R. and J. M. Tenenbaum, "Sony hack: Costs of cyber attacks and diy data management", URL http://www.logicworks.net/blog/2015/01/sony-hack-cloud-hosting-data-security/ (2015).

[53] Haghighat M., Z. S. and A.-M. M., "Cloudid: Trustworthy cloud-based and cross-enterprise biometric identification", (2015).

[54] Hoekstra, M., URL https://software.intel.com/en-us/blogs/2013/09/26/protecting-application-secrets-with-intel-sgx (2013).

[55] Hormuzd Khosravi, M. R. D. B., Alex Nayshtut, "Data protection sas v0.4", (2015).

[56] Hwang, K. and D. Li, "Trusted cloud computing with secure resources and data coloring", Internet Computing, IEEE **14**, 5, 14–22 (2010).

[57] Johnston, S., ""cloud computing" by sam johnston", URL https://commons.wikimedia.org/wiki/File:Cloud_computing.svg#/media/File:Cloud_computing.svg (2016).

[58] Kessler, G. C., *An Overview of Cryptography* (2015), URL http://www.garykessler.net/library/crypto.html#intro.

[59] Khosravi, H., "Intel identity protection technology (ipt)", URL http://csrc.nist.gov/news_events/cif_2015/trusted-computing/day2_trusted-computing_430-530.pdf (2015).

[60] kofahi, D. N. A., "An empirical study to compare the performance of some symmetric and asymmetric ciphers", International Journal of Security and Its Applications **7**, 5, 1–16 (2013).

[61] Krishnan, K., "Sfwr 4c03: Computer networks and computer security", URL http://www4.ncsu.edu/~kksivara/sfwr4c03/lectures/lecture9.pdf (2011).

[62] Krutz, R. L. and R. D. Vines, *Cloud Security: A Comprehensive Guide to Secure Cloud Computing* (Wiley Publishing, 2010).

[63] Kumar, S. N. and A. Vajpayee, "A survey on secure cloud: Security and privacy in cloud computing", American Journal of Systems and Software **4**, 1, 14–26 (2016).

[64] Lipmaa, H., D. Wagner and P. Rogaway, "Comments to nist concerning aes modes of operation: Ctr-mode encryption", (2000).

[65] Mell, P. and T. Grance, "The nist definition of cloud computing", (2011).

[66] Nate Cardozo, R. R., Kurt Opsahl, "Online service providersprivacy and transparency practices regarding government access to user data", Tech. rep., Electronic Frontier Foundation (2015).

www.manaraa.com

[67] Rijswijk-Deij, R. and E. Poll, "Using trusted execution environments in two-factor authentication: comparing approaches", (2013).

[68] Ruan, X., *Intel Identity Protection Technology: the Robust, Convenient, and Cost-Effective Way to Deter Identity Theft* (Springer, 2014).

[69] Shen, Z. and Q. Tong, "The security of cloud computing system enabled by trusted computing technology", in "Signal Processing Systems (ICSPS), 2010 2nd International Conference on", vol. 2, pp. V2–11 (IEEE, 2010).

[70] Siegler, M., "Eric schmidt: Every 2 days we create as much information as we did up to 2003", URL `"http://techcrunch.com/2010/08/04/schmidt-data/"` (2010).

[71] Solomon, M. G., *Fundamentals of Communications and Networking* (Johns and Bartlett Learning, 2014).

[72] Staff, C., URL `http://www.cio.com/article/3014617/security/data-breaches-will-affect-25-percent-of-worlds-population-by-2020-idc-predicts.html#tk.rss_all` (2015).

[73] Subashini, S. and V. Kavitha, "A survey on security issues in service delivery models of cloud computing", Journal of network and computer applications **34**, 1, 1–11 (2011).

[74] Zissis, D. and D. Lekkas, "Addressing cloud computing security issues", Future Generation computer systems **28**, 3, 583–592 (2012).

www.manaraa.com