



A SCALABLE APPROACH TO JOINT CYBER INSURANCE AND SECURITY-AS-A-SERVICE PROVISIONING IN CLOUD COMPUTING

S.P.Sivasrinivasan, Associate Professor- HOD-CSE
Department of Computer Science and Engineering,
sivzmca@gmail.com, 9444467683
E.Janitha.,- Student -M.E Computer Science and Engineering,
Sri Venkatesware Institute of Science and Technology,
Chennai.

ABSTRACT

The main aim of this project is to combined approach to security and cyber insurance provisioning in the cloud based resources. In this paper we have presented a joint approach to security and cyber insurance provisioning in the cloud. Using a stochastic optimization, we have presented a method of optimally provisioning both services in the face of uncertainty regarding future pricing, incoming traffic and cyber attacks. Thus, an application may guard against attacks by provisioning security services from providers such as Avast and Trend Micro. These services may take various forms, such as secure data storage, identity and access management (IAM), and intrusion detection services to screen incoming traffic . And then cyber insurance is used to provide explicit cover in the event that malicious activity leads to financial loss. Insurance coverage may be first- or third-party with such as theft of money and digital assets, business interruption, and cyber extortion, privacy breaches, loss of third-party data.

1 INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The project deals with the joint approach to security and cyber insurance provisioning in the cloud. Using a stochastic optimization, we have presented a method of optimally provisioning both services in the face of uncertainty regarding future pricing, incoming traffic and cyber attacks. Thus, an application may guard against attacks by provisioning security services from providers such as Avast and Trend Micro. These services may take various forms, such as secure data storage, identity and access management (IAM), and intrusion detection services to screen incoming traffic . And then cyber insurance is used to provide explicit cover in the event that malicious activity leads to financial loss. Insurance coverage may be first- or third-party with such as theft of money and digital assets, business interruption, and cyber extortion, privacy breaches, loss of third-party data.

1.2 LITERATURE SURVEY

Title: Job Dispatching and Scheduling for Heterogeneous Clusters – a Case Study on the Billing Subsystem of CHT Telecommunication

Author Name: Ting-Chou Lin

Abstract: Many enterprises or institutes are building private clouds within their own data centers. Data centers may have different batches of physical machines due to annual upgrades, but the number of machines is fixed most of the time. Consequently it is crucial to schedule jobs with different resource requirements and characteristics to meet different job timing constraints, in such heterogeneous yet most of the time static environments. This paper describes a cloud resource management framework that dynamically allocates and reallocates computation resources for jobs that have different requirements, including deadline and priority. This framework makes decisions according to specified policies, and the framework provides four default policies for system administrators to choose to fit their specific needs. The framework is designed to be componentpluggable. The components of the framework can be hotswapped, i.e., replaced without shutting down the services. In addition, the framework can work as an individual cloud computing system, or as an extension of an existing cloud system. Our experiment results demonstrate that our system is capable of dynamically adjusting the resource allocation plan according to run-time statistics collected. The system also tolerates hardware failures, and will dynamically reallocate workers to compensate for the downtime in order to finish the jobs before deadline. Our experiments also suggest a trade-off between priority and deadline.



Title: Structure, Duality, and Randomization: Common Themes in AI and OR

Author Name: Carla P. Gomes

Computer Science Department
Cornell University

Abstract: Both the Artificial Intelligence (AI) community and the Operations Research (OR) community are interested in developing techniques for solving hard combinatorial problems. OR has relied heavily on mathematical programming formulations such as integer and linear programming, while AI has developed constrained-based search and inference methods. Recently, we have seen a convergence of ideas, drawing on the individual strengths of these paradigms. Furthermore, there is a great deal of overlap in research on local search and meta-heuristics by both communities. Problem structure, duality, and randomization are overarching themes in the study of AI/OR approaches. I will compare and contrast the different views from AI and OR on these topics, highlighting potential synergistic benefits

Title: Will Cyber-Insurance Improve Network Security? A Market Analysis

Author Name: Ranjan Pal

University of Southern California

Abstract: Recent work in security has illustrated that solutions aimed at detection and elimination of security threats alone are unlikely to result in a robust cyberspace. As an orthogonal approach to mitigating security problems, some have pursued the use of cyber-insurance as a suitable risk management technique. Such an approach has the potential to jointly align with the incentives of security vendors (e.g., Symantec, Microsoft, etc.), cyber-insurers (e.g., ISPs, cloud providers, security vendors, etc.), regulatory agencies (e.g., government), and network users (individuals and organizations), in turn paving the way for comprehensive and robust cyber-security mechanisms. To this end, in this work, we are motivated by the following important question: can cyber-insurance really improve the security in a network? To address this question, we adopt a market-based approach. Specifically, we analyze regulated monopolistic and competitive cyber-insurance markets, where the market elements consist of risk-averse cyber-insurers, risk-averse network users, a regulatory agency, and security vendors. Our results show that (i) without contract discrimination amongst users, there always exists a unique market equilibrium for both market types, but the equilibrium is inefficient and does not improve network security, and (ii) in monopoly markets, contract discrimination amongst users results in a unique market equilibrium that is efficient, which in turn results in network security improvement - however, the cyber-insurer can make zero expected profits. The latter fact is often

and will eventually lead to its collapse. This fact also emphasizes the need for designing mechanisms that incentivize the insurer to permanently be part of the market

Title: Mobility Aware Task Allocation for Mobile Cloud Computing

Author Name: Bidoura Ahmad Hridita

MS Student Institute of Information

Abstract: The Mobile Cloud Computing is a promising technology that has provided a way to overcome the limitations of the mobile devices. The advancement of mobile devices technology has made the applications of these devices more complex and resource famished. Mobile cloud computing has created opportunities to execute these applications on the mobile devices by migrating the compute intensive task to the cloud. This migration of task to the cloud is not an easy task. The connectivity of the devices and the cloud is affected by the network inconsistency of wireless network. The servers on the cloud are heterogeneous in nature. Furthermore, the users are most of the time in mobile state which results in frequent change in association to access points. All of these make the selection of an optimal server to offload the task in cloud into a challenging work. In this paper, a comparative survey is provided for allocating task on the cloud along with their limitation. A mobility aware task allocation system for mobile cloud computing is also proposed. An optimization problem is formulated considering the workload and service rate of servers, network inconsistency, time to execute the task, mobility of the users etc. The proposed system aims to allocate task to the server where minimum response time is achieved in order to enhance users' quality of experience

Title: Security in Mobile Cloud Computing: A Review

Author Name: PrashantPranav

Abstract: With the implementation of cloud platforms in mobile system, the storage of bulk data by client has become easier. IT Industries are also exploiting the benefits of cloud computing by producing more and more smart phones that takes full benefit of the features of clouds. As the use of smart phones by users is increasing rapidly, the issue of security related to use of cloud computing technique in mobile computing environment has emerged as one of the biggest challenges in this regard. Security with respect to mobile cloud computing can be addressed at three levels viz. mobile terminal, mobile network security, and cloud storage. Although many attempts have been made in developing a model which ensures privacy and security of data in mobile cloud system, no model is free from malicious attacks. In this review paper, we have focused on few models which are



(VANETs), because of the nonexistence of end-to-end connections, it is essential that nodes take advantage of connection opportunities to forward messages to make end-to-end messaging possible. Thus, it is crucial to make sure that nodes have incentives to forward messages for others, despite the fact that the routing protocols in VANETs are different from traditional end-to-end routing protocols. In this paper, stimulation of message forwarding in VANETs is concerned. This approach is based on coalitional game theory, particularly, an incentive scheme for VANETs is proposed and with this scheme, following the routing protocol is in the best interest of each node. In addition, a lightweight approach is proposed for taking the limited storage space of each node into consideration.

2 SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

In the previous systems have been maintain the security services only.so service should be protect the user system and information .Sometimes may attacker attack the user assets and data.like incidents and disasters such as data breach, data corruption, and business interruption. However, one successful attack can result in the loss of data and revenue worth millions of dollars. that compensate them for customers may also purchase cyber insurance to receive recompense in the case of loss.so loss will be uncertain. We cannot assume that damages can be accurately determined. This may be in various forms, such as a 'ransom' paid . It is necessary to balance provisioning of security and insurance, even when future costs and risks are uncertain. One of the key challenges in cyber insurance is the accurate estimation of damages caused by cyber attacks

DISADVANTAGE:

- Single server Management.
- Increase cyber attack to unpredictable customer datas.

2.2 PROPOSED SYSTEM

In this paper, we present SECaaS in firewall-style to provide a security policy enforcement and monitoring infrastructure for network traffic. which focuses on network traffic analysis like IDS (Intrusion Detection System) implementations to identify attack behaviours. And then relationship between cyber insurance and SECaaS provisioning, containing a customer who uses applications, which receive Internet traffic in the form of packets. These packets are scanned by services from SECaaS providers, provisioned by a subscription management process (SMP). In

subscribed to by an insurance management process (IMP), provide compensation for damages incurred. In this application run on customer machine that we assume to be Internet-accessible, either on a cloud service such as Amazon. Applications receive data packets in accordance with their operating purpose, e.g. email data or financial transactions. Legitimate packets are called safe packets, while packets used in cyber attacks are called unsafe packets. Unsafe packets are deemed handled if they are correctly detected by security services, or unhandled if they are not successfully processed (for example if they are undetected). These unhandled packets will cause damage, which incurs costs to the customer will refund the amount to insurance company. And then IMP will refund the particular data cost to customer.

ADVANTAGE:

- User gets security.
- Provisioning both services in the face of uncertainty regarding future pricing, incoming traffic and cyber attacks.

2.3 SYSTEM REQUIREMENTS

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

2.3.1 HARDWARE REQUIREMENTS

- Hard disk: 500 GB and above.
- Processor: i3 and above.
- Ram : 4GB and above

2.3.2 SOFTWARE REQUIREMENTS

- Operating System: Windows 7 and above (64-bit).
- Java Version : JDK 1.7
- Web Server : Tomcat 6.20
- Web Server : Tomcat 7.0.11

2.4 TECHNOLOGIES USED

a) JAVA

It is a Platform Independent. Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C.



INTRODUCTION TO JAVA

Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project. The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak.

The first implementation of Oak was in a PDA-type device called Star Seven (*7) that consisted of the Oak language, an operating system called GreenOS, a user interface, and hardware. The name *7 was derived from the telephone sequence that was used in the team's office and that was dialed in order to answer any ringing telephone from any other phone in the office.

Around the time the First Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World Wide Web, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses. The reason for this was the introduction of a graphical Web browser called Mosaic, developed by ncSA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic.

It has only been since 1994 that Oak technology has been applied to the Web. In 1994, two Sun developers created the first version of Hot Java, and then called Web Runner, which is a graphical browser for the Web that exists today. The browser was coded entirely in the Oak language, by this time called Java. Soon after, the Java compiler was rewritten in the Java language from its original C code, thus proving that Java could be used effectively as an application language. Sun introduced Java in May 1995 at the Sun World 95 convention.

Web surfing has become an enormously popular practice among millions of computer users. Until Java, however, the content of information on the Internet has been a bland series of HTML documents. Web users are hungry for applications that are interactive, that users can execute no matter what hardware or software platform they are using, and that travel across heterogeneous networks and do not spread viruses to their computers. Java can create such applications.

a) WORKING OF JAVA

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data (called attributes) and functions (called methods).

When the interpreter executes a class, it looks for a particular method by the name of main, which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the argv[] of C), and is declared as a static method.

To output text from the program, we execute the println method of System.out, which is java's output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Windows platform, it will write the string passed to it to the user's program.

Java consists of two things :

- Programming language
- Platform

b) Cloud Computing

- Cloud computing is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet.
- Users need not have knowledge of, expertise in, or control over the technology infrastructure in the "cloud" that supports them.
- The concept generally incorporates combinations of the following: • infrastructure as a service (IaaS) • platform as a service (PaaS) • software as a service (SaaS)
- Cloud computing customers do not generally own the physical infrastructure serving as host to the software platform in question. Instead, they avoid capital expenditure by renting usage from a third-party provider.
- They consume resources as a service and pay only for resources that they use.
- Many cloud-computing offerings employ the utility computing model, which is analogous to how traditional utility services (such as electricity) are consumed.

c) THE JAVA PROGRAMMING LANGUAGE

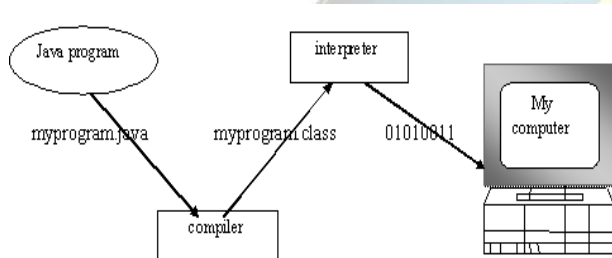
Java is a high-level programming language that is all of the following:



- Distributed
- Interpreted
- Robust
- Secure
- Architecture-neutral
- Portable
- High-performance
- Multithreaded
- Dynamic

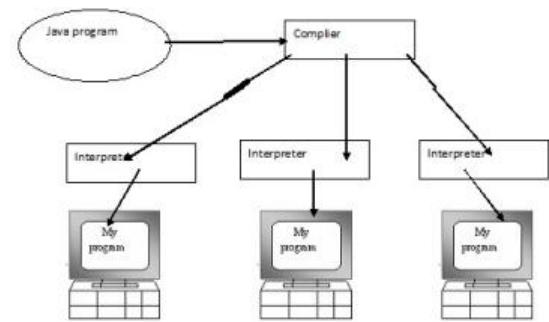
The code can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Java is unusual in that each Java program is both compiled and interpreted. With a compiler, you translate a Java program into an intermediate language called **Java byte codes** – the platform independent codes interpreted by the Java interpreter. With an interpreter, each Java byte code instruction is parsed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. This figure illustrates how it works :



You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (JVM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of JVM. That JVM can also be implemented in hardware. Java byte codes help make “write once, run anywhere” possible.

You can compile your Java program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the JVM. For example, that same Java program can be run on Windows NT, Solaris and Macintosh



Windows NT

System

8

d) THE JAVA PLATFORM

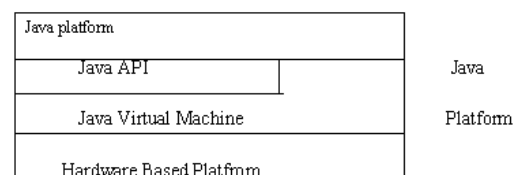
A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system.

The Java platform has two components :

- The Java Virtual Machine (JVM)
- The Java Application Programming Interface (Java API)

You've already been introduced to the JVM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries (**packages**) of related components. The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.



As a platform-independent environment, Java can be a bit slower than native code. However, smart compilers,



bring Java's performance close to that of native code without threatening portability.

WORKING OF JAVA

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data and functions.

When the interpreter executes a class, it looks for a particular method by the name of main, which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the argv[] of C), and is declared as a static method.

To output text from the program, we execute the println method of System.out, which is java's output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Wintel platform, it will write the string passed to it to the user's program.

APACHE TOMCAT SERVER

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

Environment

Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets.

The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments.

Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

Product features

- Tomcat 3.x (initial release)
- Implements the Servlet 2.2 and JSP 1.1 specifications Servlet reloading
- Basic HTTP functionality Tomcat 4.x
- Implements the Servlet 2.3 and JSP 1.2 specifications
- Servlet container redesigned as Catalina
- JSP engine redesigned as Jasper
- Coyote connectorJava Management Extensions (JMX),JSP&Struts-basedadministrationTomcat 5.x
- Implements the Servlet 2.4 and JSP 2.0 specifications
- Reduced garbage collection, improved performance and scalability
- Native Windows and Unix wrappers for platform integration
- Faster JSP parsing

History

Tomcat started off as a servlet specification implementation by James Duncan Davidson, a software architect at Sun. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation.

Davidson had initially hoped that the project would become open-sourced and, since most open-source projects had O'Reilly books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with Tomcat since he reasoned the animal represented something that could take care of and fend for itself. His wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a tomcat on the cover.

Introduction

Purpose

The main aim of this project is to combined approach to security and cyber insurance provisioning in the cloud based resources.

Project Scope

In this paper we have presented a joint approach to security and cyber insurance provisioning in the cloud. Using a stochastic optimization, we have presented a method of optimally provisioning both services in the face of uncertainty regarding future pricing, incoming traffic and cyber attacks. Thus, an application may guard against attacks by provisioning security services from providers such as Avast and Trend Micro. These services may take various forms, such as secure data storage, identity and access management (IAM), and intrusion detection services to screen incoming traffic . And then cyber insurance is used to provide explicit cover in the event that malicious activity leads to financial loss. Insurance coverage may be first- or third-party with such as theft of money and digital assets, business interruption, and



Overall Description

Product Perspective

In the previous systems have been maintain the security services only.so service should be protect the user system and information .Sometimes may attacker attack the user assets and data.like incidents and disasters such as data breach, data corruption, and business interruption. However, one successful attack can result in the loss of data and revenue worth millions of dollars. that compensate them for customers may also purchase cyber insurance to receive recompense in the case of loss.so loss will be uncertain. We cannot assume that damages can be accurately determined. This may be in various forms, such as a 'ransom' paid . It is necessary to balance provisioning of security and insurance, even when future costs and risks are uncertain. One of the key challenges in cyber insurance is the accurate estimation of damages caused by cyber attacks

Product Features

In this paper, we present SECaaS in firewall-style to provide a security policy enforcement and monitoring infrastructure for network traffic. which focuses on network traffic analysis like IDS (Intrusion Detection System) implementations to identify attack behaviours. And then relationship between cyber insurance and SECaaS provisioning, containing a customer who uses applications, which receive Internet traffic in the form of packets. These packets are scanned by services from SECaaS providers, provisioned by a subscription management process (SMP). In the event that harmful packets elude security, cyber insurers, subscribed to by an insurance management process (IMP), provide compensation for damages incurred. In this application run on customer machine that we assume to be Internet-accessible, either on a cloud service such as Amazon. Applications receive data packets in accordance with their operating purpose, e.g. email data or financial transactions. Legitimate packets are called safe packets, while packets used in cyber attacks are called unsafe packets. Unsafe packets are deemed handled if they are correctly detected by security services,or unhandled if they are not successfully processed (for example if they are undetected). These unhandled packets will cause damage, which incurs costs to the customer will refund the amount to insurance compnay.And then IMP will refund the particular data cost to customer.

User Classes and Characteristics

1. Purchase the Security services:

In this module user first register the cloud site and the provide user details (Name,password,email,,mobile,dob)

username,password.Once user name and password is valid open the user profile screen will be display.After login user will purchase the outsource security service in cloud.In the security service have a various control and price,validity.User will choose our system performance based services and then immediately transfer amount to security management.

Once got the service , will protect the customer application and system to particular time periods.

2. Cloud service:

In the module, user register the cloud service based on user credential details and then login the cloud resource.Once enter the cloud site or application to utilize the site.Ifyour application may be social network to share your post and chat with our friends. Users will upload their pictures into the social networking site. While uploading, user provides tags for the picture At the same time security system will protect the application to each and every request to cloud And then way of securing cloud-based data.

3 . Screening Data traffic:

In our security model ,service managed by the customer applications and then monitor the traffic flow and screening incoming data packets in accordance with their operating purpose, e.g. email data or financial transactions,webpages Legitimate packets are called safe packets, while packets used in cyber attacks are called unsafe packets. Unsafe packets are deemed handled if they are correctly detected by security services, or unhandled if they are not successfully processed (for example if they are undetected). These unhandled packets will cause damage, which incurs costs to the customer.so SECaaS to noted on user packet size.Atthe same time service will redirect to insurance management process(IMP).

4 . Claim Insurance:

In this module IMP will check the user if customer or not .And then check the customer current premium data And evaluate the current unhandled data size to calculate the particular per-packet,price, duration, and maximum number of packets affected. we introduce a partial Lagrange multiplier algorithm to find the optimal solution in parameter change to calculate the amount to data size .And then refund the amount to particular customer. After claim the customer current premium is low to change the new future premium based on incoming unsafe packets. The price for insurance purchased in advance is charged at a rate known as a 'future premium'. The IMP purchases insurance policies, which includes the premium, types of risks covered, indemnity value, and policy duration.



a) Design and Implementation Constraints

Constraints in Analysis

- Constraints as Informal Text
- Constraints as Operational Restrictions
- Constraints Integrated in Existing Model Concepts
- Constraints as a Separate Concept
- Constraints Implied by the Model Structure

Constraints in Design

- Determination of the Involved Classes
- Determination of the Involved Objects
- Determination of the Involved Actions
- Determination of the Require Clauses
- Global actions and Constraint Realization

Constraints in Implementation

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

External Interface Requirements

a) User Interfaces

1. All the contents in the project are implemented using Graphical User Interface (GUI) in Java through JavaFX concepts.
2. Every conceptual part of the projects is reflected using the JavaFX.
3. System gets the input and delivers through the GUI based.

b) Hardware Interfaces

Ethernet

Ethernet on the AS/400 supports TCP/IP, Advanced Peer-to-Peer Networking (APPN) and advanced program-to-program communications (APPC).

ISDN

You can connect your AS/400 to an Integrated Services Digital Network (ISDN) for faster, more accurate data transmission. An ISDN is a public or private digital communications network that can support data, fax, image, and other services over the same physical interface. Also, you can use other protocols on ISDN, such as IDLC and X.25.

c) Software Interfaces

This software is interacted with the TCP/IP protocol, Socket and listening on unused ports. Server Socket and listening on unused ports and JDK 1.6

d) Communications Interfaces

1. TCP/IP protocol.
2. LAN settings

1. Other Nonfunctional Requirements Performance Requirements

The performance of the wireless sensor network, to execute this project on LAN or wifi communication channel. So we need to one or more than machine to execute the demo. Machine needs the enough hard disk space to install the software and run our project.

1.2 Safety Requirements

- The software may be safety-critical. If so, there are issues associated with its integrity level
- The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
- If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
- There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
- If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
- Systems with different requirements for safety levels must be separated.
- Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

Security Requirements

Do not block the some available ports through the windows firewall

Software Quality Attributes

Functionality: are the required functions available, including Interoperability and security.

Reliability: maturity, fault tolerance and recoverability

Usability: how easy it is to understand, learn, and operate the software System

Efficiency: performance and resource behavior.

Maintainability: Maintaining the software.

Portability: can the software easily be transferred to another environment, Including installability

Fig 3.1 Sequence Diagram:

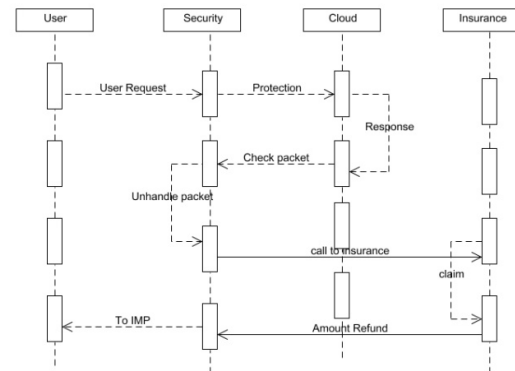


Fig 3.2 Use Case Diagram:

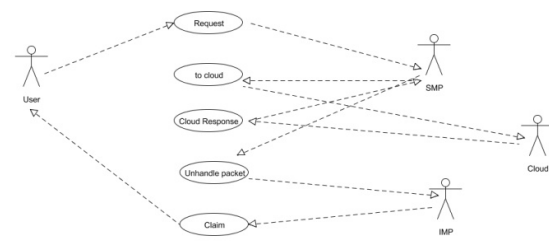


Fig 3.3 Activity Diagram:

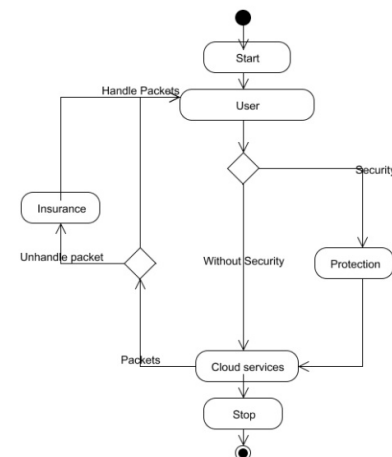
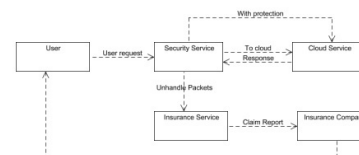


Fig 3.4 Collaboration Diagram:



3 SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

Architecture

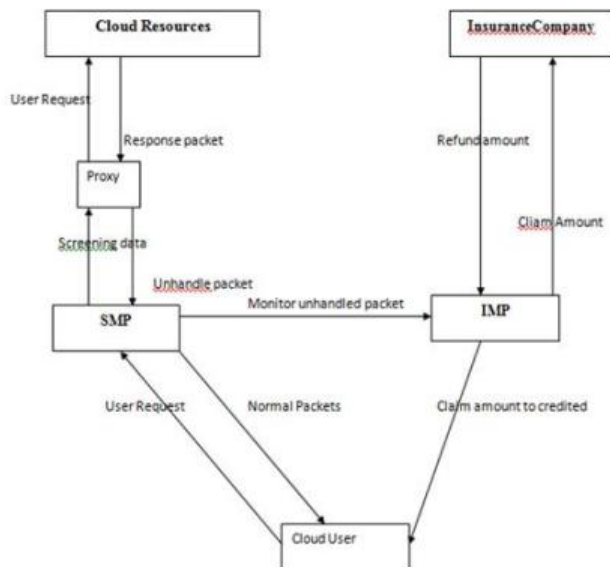
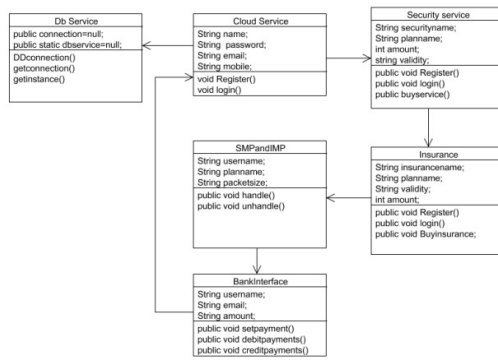
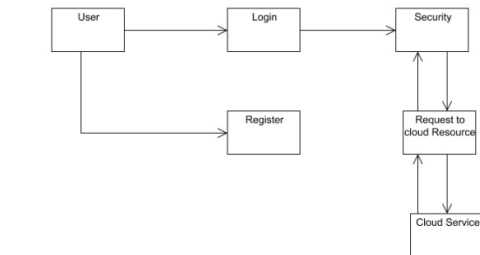




Fig 3.5 Class Diagram:



Level 2:



Level 3:

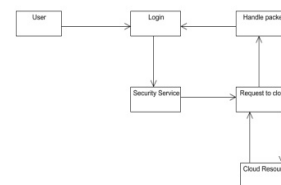
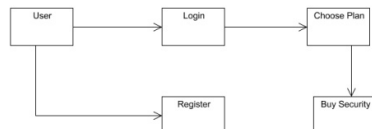


Fig 3.6 Data Flow Diagram:

Level 0:



Level 1:

3.2 SYSTEM DESIGN

3.2.1MODULES

- ✓ Purchase the Security services.
- ✓ Cloud service.
- ✓ Screening Data traffic.
- ✓ Claim Insurance



Module Description:

1. Purchase the Security services.

In this module user first register the cloud site and the provide user details (Name,password,email,,mobile,dob) .And then login the user credential details like username,password.Once user name and password is valid open the user profile screen will be display.After login user will purchase the outsource security service in cloud.In the security service have a various control and price,validity.User will choose our system performance based services and then immediately transfer amount to security management.

Once got the service , will protect the customer application and system to particular time periods.

2. Cloud service.

In the module, user register the cloud service based on user credential details and then login the cloud resource.Once enter the cloud site or application to utilize the site.Ifyour application may be social network to share your post and chat with our friends. Users will upload their pictures into the social networking site. While uploading, user provides tags for the picture At the same time security system will protect the application to each and every request to cloud And then way of securing cloud-based data.

3. Screening Data traffic.

In our security model ,service managed by the customer applications and then monitor the traffic flow and screening incoming data packets in accordance with their operating purpose, e.g. email data or financial transactions,webpages Legitimate packets are called safe packets, while packets used in cyber attacks are called unsafe packets. Unsafe packets are deemed handled if they are correctly detected by security services, or unhandled if they are not successfully processed (for example if they are undetected). These unhandled packets will cause damage, which incurs costs to the customer.so SECaaS to noted on user packet size.At the same time service will redirect to insurance management process(IMP).

4. Claim Insurance

In this module IMP will check the user if customer or not .And then check the customer current premium data And evaluate the current unhandled data size to calculate the particular per-packet,price, duration, and maximum number of packets affected. we introduce a partial Lagrange multiplier algorithm to find the optimal solution in parameter change to calculate the amount to data size .And then refund the amount to particular customer. After claim the customer current premium is low to change the new future premium based on incoming unsafe packets. The price for insurance purchased in advance is charged at a rate known as a 'future premium'. The

premium, types of risks covered, indemnity value, and policy duration.

4 CODING AND TESTING

4.1 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

- Program should be simple, clear and easy to understand.
- Naming conventions
- Value conventions
- Script and comment procedure
- Message box format
- Exception and error handling

4.1.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The conventions are adopted for **easy understanding** of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

4.1.2 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:



- Proper default values for the variables.
- Proper validation of values in the field.
- Proper documentation of flag values.

4.1.3 SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

4.1.4 MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

- X – User has performed illegal operation.
- ! – Information to the user.

4.2 TEST PROCEDURE

SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

4.3 TEST DATA AND OUTPUT

4.3.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

4.3.2 FUNCTIONAL TEST

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

4.3.3 PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

4.3.4 STRESS TEST

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

4.3.5 STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling end of file condition, I/O errors, buffer problems and textual errors in output information

4.3.6 INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time



i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

4.3.7 TESTING TECHNIQUES / TESTING STRATEGIES

a) TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

b) WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

c) BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

d) SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.



e) INTEGRATION TESTING:

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”-interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

f) PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

g) SECURITY TESTING

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

h) VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation

- The function or performance characteristics confirm to specifications and are accepted.
- A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic.

i) USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

- Input screen design.
- Output screen design.

5CONCLUSION

In this project we provide a combined approach to security and cyber insurance provisioning in the cloud. Using a stochastic optimization, we have presented a method of optimally provisioning both services in the face of uncertainty regarding future pricing, incoming traffic and cyber attacks. We predict Accuracy of data could further be extended through the implementation of systems to update parameters on a daily or weekly basis, to improve future Decisions.

Future work:

In the future work we will approach the real honeypot data prediction in joint security and insurance provisioning. The real honeypot data provided by the University ofWaikato’s Cyber Security Lab . Honeypots positioned in Singapore, Sao P’aolo, Brazil, and San Jose, USA, collected packet data over a number of days. where the security performance of one part of the system can impact the security of other parts.



APPENDIX

SOURCE CODE

LOGIN

```
package Logic;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Properties;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class Login extends HttpServlet {

    /**
     * The doGet method of the servlet. <br>
     *
     * This method is called when a form has its tag
     * value method equals to get.
     *
     * @param request the request send by the client to
     * the server
     * @param response the response send by the server
     * to the client
     * @throws ServletException if an error occurred
     */
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException,
        IOException {
        try {
            HttpSession hs=request.getSession(true);
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            String name=request.getParameter("name");
            String
            pass=request.getParameter("password");
            System.out.println("-----"
            "+name+pass");
            Connection
            con=(Connection)getServletContext().getAttribute("Connectio
            n");
            PreparedStatement pr=con.prepareStatement("select *
            from securityregister where username=? and password=?");
            pr.setString(1, name);
            pr.setString(2, pass);
            ResultSets=pr.executeQuery();
            if(rs.next())
            {
                hs.setAttribute("Username",
                name);

                RequestDispatcher rd=request.getRequestDispatcher(
                "Home.jsp");

                rd.forward(request, response);
            }
        }
        catch (Exception e) {
            out.println(e.getMessage());
        }
    }
}
```



```
{  
  
    RequestDispatcher rd=request.getRequestDispatcher(  
"index.jsp");  
  
    rd.forward(request, response);  
  
}  
  
}  
  
catch (Exception e) {  
    e.printStackTrace();  
}  
  
}
```

REGISTER

```
package Logic;  
  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import javax.servlet.RequestDispatcher;  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
public class Register extends HttpServlet {  
  
    public void doGet(HttpServletRequest request,  
HttpServletResponse response)  
  
        throws ServletException,  
IOException  
  
    {  
  
        try
```

```
        response.setContentType("text/html");  
  
        PrintWriter out = response.getWriter();  
  
        String name=request.getParameter("name");  
  
        String  
pass=request.getParameter("password");  
  
        String mob=request.getParameter("phone");  
  
        String email=request.getParameter("email");  
  
        System.out.println("-----  
"+name+pass+mob+email);  
  
        Connection  
con=(Connection)getServletContext().getAttribute("Conne  
ction");  
  
        PreparedStatement pr=con.prepareStatement("insert  
into securityregister (username,password,mobile,email)  
values(?,?,?,?)");  
  
        pr.setString(1, name);  
        pr.setString(2, pass);  
        pr.setString(3, mob);  
        pr.setString(4, email);  
        pr.executeUpdate();  
  
        RequestDispatcher rd=request.getRequestDispatcher(  
"index.jsp");  
  
        rd.forward(request, response);  
  
    }  
  
    catch (Exception e)  
  
    {  
  
        e.printStackTrace();  
  
    }  
  
    }  
  
}
```

REFERENCES

- [1] McAfee, "Net Losses: Estimating the Global Cost of Cybercrime." Center for Strategic and International



- Studies, Economic Impact of Cybercrime II, Jun. 2014.
- [2] (2016) Identity theft resource center data breach reports. [Online]. Available: <http://www.idtheftcenter.org/2016databreaches.html>
- [3] (2016) A bold approach to cyber risk management. [Online]. Available: <http://www.mas.gov.sg/News-and-Publications/Speechesand-Monetary-Policy-Statements/Speeches/2016/A-Bold-Approach-to-Cyber-Risk-Management.aspx>.
- [4] (2016) Insurance 2020 beyond: Reaping the dividends of cyber resilience. [Online]. Available: <http://www.pwc.com/gx/en/industries/financialservices/insurance/publications/insurance-2020-cyber.html>
- [5] (2016) McAfee security-as-a-service solutions. [Online]. Available: <https://www.mcafeeasap.com/MarketingContent/Products/ProductsLanding.aspx>
- [6] (2016) Deep security as a service. [Online]. Available: <http://www.trendmicro.com/us/business/saas/deep-security-as-a-service/#usage-based-pricing>
- [7] (2016) Allianz cyber protect. [Online]. Available: <http://www.agcs.allianz.com/services/financial-lines/cyber-insurance/>
- [8] Christo Ananth, Kavya.S., Karthika.K., Lakshmi Priya.G., Mary Varsha Peter, Priya.M., "CGT Method of Message forwarding", International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE), Volume 1, Issue 1, August 2015, pp:10-15
- [9] M. Clark. (2014) Timeline of target's data breach and aftermath: How cybertheft snowballed for the giant retailer. [Online]. Available: <http://www.ibtimes.com/timeline-targets-data-breachaftermath-how-cybertheft-snowballed-giant-retailer-1580056>
- [10] C. A. Newman. (2016) Targets cyber insurance: A \$100 million policy vs. \$300 million (so far) in costs. [Online]. Available: <http://datasecuritylaw.com/blog/targets-cyber-insurancea->
- [11] 100-million-policy-vs-300-million-so-far-in-costs/ S. optimization approach to security-as-a-service allocation and cyber insurance
- [12] management," in Trustcom/BigDataSE/ISPA, 2015 IEEE, Aug 2015, pp. 426–433.
- [13] C. P. Ram and G. Sreenivaasan, "Security as a service (sass): Securing user data by coprocessor and distributing the data," in Trendz in Information Sciences Computing(TISC2010), Dec 2010, pp.152–155.