Performance evaluation of security mechanisms in Cloud Networks

ANAND KANNAN

KTH Information and Communication Technology

Degree project in
Communication Systems
Second level, 30.0 HEC
Stockholm, Sweden

Performance evaluation of security mechanisms in Cloud Networks

Anand Kannan

30th July - 2012

Supervisors:

Prof. Gerald Q. Maguire Jr.	Dr.Volker Fusenig & Mr. Ayush Sharma
KTH Royal Institute of Technology	Fraunhofer AISEC,
Stockholm, Sweden	Munich, Germany

Abstract

Infrastructure as a Service (IaaS) is a cloud service provisioning model which largely focuses on data centre provisioning of computing and storage facilities. The networking aspects of IaaS beyond the data centre are a limiting factor preventing communication services that are sensitive to network characteristics from adopting this approach. Cloud networking is a new technology which integrates network provisioning with the existing cloud service provisioning models thereby completing the cloud computing picture by addressing the networking aspects. In cloud networking, shared network resources are virtualized, and provisioned to customers and end-users on-demand in an elastic fashion. This technology allows various kinds of optimization, e.g., reducing latency and network load. Further, this allows service providers to provision network performance guarantees as a part of their service offering. However, this new approach introduces new security challenges. Many of these security challenges are addressed in the CloNe security architecture.

This thesis presents a set of potential techniques for securing different resource in a cloud network environment which are **not** addressed in the existing CloNe security architecture. The thesis begins with a holistic view of the Cloud networking, as described in the Scalable and Adaptive Internet Solutions (SAIL) project, along with its proposed architecture and security goals. This is followed by an overview of the problems that need to be solved and some of the different methods that can be applied to solve parts of the overall problem, specifically a comprehensive, tightly integrated, and multi-level security architecture, a key management algorithm to support the access control mechanism, and an intrusion detection mechanism. For each method or set of methods, the respective state of the art is presented. Additionally, experiments to understand the performance of these mechanisms are evaluated on a simple cloud network test bed.

The proposed key management scheme uses a hierarchical key management approach that provides fast and secure key update when member join and member leave operations are carried out. Experiments show that the proposed key management scheme enhances the security and increases the availability and integrity.

A newly proposed genetic algorithm based feature selection technique has been employed for effective feature selection. Fuzzy SVM has been used on the data set for effective classification. Experiments have shown that the proposed genetic based feature selection algorithm reduces the number of features and hence decreases the classification time, while improving detection accuracy of the fuzzy SVM classifier by minimizing the conflicting rules that may confuse the classifier. The main advantages of this intrusion detection system are the reduction in false positives and increased security.

Sammanfattning

Infrastructure as a Service (IaaS) är en Cloudtjänstmodell som huvudsakligen är inriktat på att tillhandahålla ett datacenter för behandling och lagring av data. Nätverksaspekterna av en cloudbaserad infrastruktur som en tjänst utanför datacentret utgör en begränsande faktor som förhindrar känsliga kommunikationstjänster från att anamma denna teknik. Cloudnätverk är en ny teknik som integrerar nätverkstillgång med befintliga cloudtjänstmodeller och därmed fullbordar föreställningen av cloud data genom att ta itu med nätverkaspekten. I cloudnätverk virtualiseras delade nätverksresurser, de avsätts till kunder och slutanvändare vid efterfrågan på ett flexibelt sätt. Denna teknik tillåter olika typer av möjligheter, t.ex. att minska latens och belastningen på nätet. Vidare ger detta tjänsteleverantörer ett sätt att tillhandahålla garantier för nätverksprestandan som en del av deras tjänsteutbud. Men denna nya strategi introducerar nya säkerhetsutmaningar, exempelvis VM migration genom offentligt nätverk. Många av dessa säkerhetsutmaningar behandlas i CloNe's Security Architecture. Denna rapport presenterar en rad av potentiella tekniker för att säkra olika resurser i en cloudbaserad nätverksmiljö som inte behandlas i den redan existerande CloNe Security Architecture.

Rapporten inleds med en helhetssyn på cloudbaserad nätverk som beskrivs i Scalable and Adaptive Internet Solutions (SAIL)-projektet, tillsammans med dess föreslagna arkitektur och säkerhetsmål. Detta följs av en översikt över de problem som måste lösas och några av de olika metoder som kan tillämpas för att lösa delar av det övergripande problemet. Speciellt behandlas en omfattande och tätt integrerad multi-säkerhetsarkitektur, en nyckelhanteringsalgoritm som stödjer mekanismens åtkomstkontroll och en mekanism för intrångsdetektering. För varje metod eller för varje uppsättning av metoder, presenteras ståndpunkten för respektive teknik. Dessutom har experimenten för att förstå prestandan av dessa mekanismer utvärderats på testbädd av ett enkelt cloudnätverk.

Den föreslagna nyckelhantering system använder en hierarkisk nyckelhantering strategi som ger snabb och säker viktig uppdatering när medlemmar ansluta sig till och medlemmarna lämnar utförs. Försöksresultat visar att den föreslagna nyckelhantering system ökar säkerheten och ökar tillgänglighet och integritet.

En nyligen föreslagna genetisk algoritm baserad funktion valet teknik har använts för effektiv funktion val. Fuzzy SVM har använts på de uppgifter som för effektiv klassificering. Försök har visat att den föreslagna genetiska baserad funktion selekteringsalgoritmen minskar antalet funktioner och därmed minskar klassificering tiden, och samtidigt förbättra upptäckt noggrannhet fuzzy SVM klassificeraren genom att minimera de motstående regler som kan förvirra klassificeraren. De främsta fördelarna med detta intrångsdetekteringssystem är den minskning av falska positiva och ökad säkerhet.

Acknowledgements

My sincere thanks are due to my thesis supervisor Prof. Dr. Gerald Q. “Chip” Maguire Jr. (School of Information and Communication Technologies, KTH Royal Institute of Technology, Stockholm, Sweden) for his valuable suggestions, thought provoking ideas, and indispensable recommendations. I am very grateful for his spending valuable time in guiding me and getting back to me in a very short span whenever approached.

My heartfelt gratitude to my thesis advisors Mr. Ayush Sharma and Dr.Volker Fusening (Network security and early warning systems, Fraunhofer, AISEC), Munich, Germany) for sharing their valuable ideas and also personally helping me settle down in a new country and successfully complete my work in a very short span.

Special thanks to Mr. Peter Schoo (Network security & early warning systems, Fraunhofer AISEC) for his constant support he gave me both technically and personally and also for giving me opportunity to carry out my thesis at Fraunhofer’s facility during my tenure at Fraunhofer, AISEC.

My gratitude to Ms. May-Britt Eklund_Larsson for the continuous support she lend during my course work days at KTH, Stockholm, Sweden. I also appreciate the constant support of the group members of Fraunhofer AISEC.

Furthermore I would like to thank Dr. Vijay Kumar and Mr. Ganapathy Sannasi (Anna University Chennai) for spending their valuable time in guiding me during my Master’s thesis work.

Finally, I thank my parents for their uninterrupted affection and moral support throughout the period of my study, and all through my life. I would like to thank my special one, friends, family members, and everyone else who supported and inspired me during my whole life.

25-July-2012, Stockholm.
Anand Kannan

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACL	Access Control List
CA	Certification authority
CIDN	Collaborative intrusion detection network
CloNe	Cloud network
DCHK	Date-constraint hierarchical key
DCP	Distributed control plane
DoS	Denial of service
EC2	Elastic compute cloud
FAR	False alarm rate
FDT	Fuzzy decision tree
GbE	Gigabit Ethernet
GRC	Governance, Risk Management & Compliance
IPSec	Internet protocol security
IaaS	Infrastructure as a service
IT	Information technology
KPI	Key performance indicator
LAN	Local area network
LAG	Link aggregation group
MA	Mobile agents
QoS	Quality of service
R2L	Remote to Local (attack)
SAIL	Scalable and Adaptive Internet Solutions
SLA	Service level agreement
TRILL	Transparent Interconnection of Lots of Links
U2R	User to Root (attack)
VLAN	Virtual local area network
VM	Virtual machine
VoIP	Voice over Internet Protocol
VRRP	Virtual Router Redundancy Protocol
VxLAN	Virtual eXtensible local area network
WAN	Wide area network

1 Introduction

Cloud computing has gained remarkable popularity in the recent years among a wide range of consumers, ranging from small start-ups to multinational companies. The advantages of deploying and running applications in the cloud are manifold: lower costs through use of shared computing resources, no upfront infrastructure costs, and on-demand provisioning of computing nodes to fit transient requirements. Thus, for applications that show a high degree of variable demand for resources, the cloud computing model offers an efficient and cost-effective method to provide the resources used, while minimizing economic cost. Virtualization within data centres has been a key enabler by allowing the dynamic provisioning of computing resources. However, virtualization benefits are limited by poor network flexibility which can lead to underutilization of computing resources during peak loads. It is obvious that the perceived performance of most applications running in the cloud should depend heavily on the network connections between both the different cloud sites and between the users and the cloud. [3]

Applications with interactive and bandwidth hungry characteristics are a good example of the applications which face problems due to their network communications. As these applications increasingly move to the cloud, more will be demanded from existing networks in terms of better service, for example capacity (as it is likely more data that will be sent across network links), quality (low delay for interactive applications), and availability. Cloud applications may demand a network that is more flexible, since applications and entire clusters of servers can be moved to another data centre and hence existing networking pipes need to be re-plumbed. Existing technology provides the allocation of computing resources in the cloud in a dynamic and rapid fashion, but the network connections to these resources are more or less statically configured by network operators [3]. The SAIL project addresses cloud networking as a combination of management for cloud computing and managing the vital networking capabilities between distributed cloud resources in order to improve the management of both. Since Cloud Networking integrates cloud computing deeply into networks, for efficient network operations and service awareness, it must provide on-demand guaranteed network resources within a time span that is compatible with the allocation of computing resources in a cloud today [2].

1.1 Central goals of this thesis project

This thesis project is primarily focused on Cloud Networking Security. The project has the following goals:

- Extend and concretize the already existing (albeit high-level) security architecture proposed for SAIL, and to compare the CloNe architecture with existing cloud security architectures.
- Propose an enhanced distributed key management protocol and evaluate it in a simple cloud test bed.
- Propose an enhanced intrusion detection system and evaluate it in a simple cloud test bed.

1.2 Organisation of this report

This report is organised as follows Chapter 2 presents the basic concepts of cloud networking and the virtualization technologies supporting cloud networking. This is followed by a description of the Cloud Networking (CloNe) architecture in more detail, with special emphasis on the roles, interfaces, distinction between single and cross-domain infrastructures, and the resources involved in the CloNe provisioning model. This chapter also elaborates on the core networking technologies in CloNe, and the Flash Network Slice (FNS) concept, which has been introduced as a part of the CloNe paradigm. Chapter 2 concludes by describing the security challenges in a cloud networking environment. Chapter 3 starts with the related work regarding the different components which would be developed as a part of the thesis work. The chapter starts with the related work regarding existing security architectures and their comparison with the CloNe security architecture. This is followed by related work in access control mechanisms and key management methodologies, with the focus on the security goal translation for cloud networks. Chapter 4 gives a detailed description of the proposed architecture. Chapter 5 describes the implementation, followed by testing of the implemented module in chapter 6. Chapter 7 analyses the proposed key management mechanism. Chapter 8 gives an overview of the Intrusion Detection module - which is part of this thesis project and was submitted as a conference paper. (The paper will be included as an appendix.) Chapter 9 presents some conclusions from this work and suggests some future work.

2 Concepts of Cloud Networking Technology

This chapter provides some background regarding the virtualized network resource provisioning model, which is the core of the CloNe architecture. Moreover, the chapter also covers the virtualization technologies used to realize the architecture. More specifically, details are provided regarding the CloNe architecture, core technologies used to implement CloNe, the FNS concept, and flow-based networking.

2.1 Virtualized network resource provisioning

Today’s implementation of cloud infrastructures is built on server virtualization [4] [5] [6], network virtualization (programmable transport networks [7] [8] [9] [10]), and storage virtualization like Amazon’s Elastic Block Store [11]. Infrastructure as a service (IaaS) management systems deploy and manage virtual machines, networks, and data stores upon demand by the customer, thus enabling a dynamically changeable infrastructure topology. These virtualization techniques are in such demand that standard server chip sets by vendors such as Intel included technology to improve the efficiency of virtualization (VT-x [12]).

With continuing economic uncertainty and high levels of business risk, enterprises have focused on flexibility and renewed business agility. According to Forrester research [14], IaaS is the area of cloud computing that currently receives the most market attention- with more than 26% of enterprises planning to adopt IaaS via an external provider. The IaaS business model drives infrastructure providers towards a centralised architecture with deployment of very large data centres that optimize a combination of low cost in land, power, and labour. This combination results in the lowest cost for the provider. However, the business requirements bring in opposite factors. From a regulatory perspective, the data centre’s location determines in part the legal jurisdiction that applies to hosted services (e.g., USA Patriot Act [11]). The use of the services can restrict their location or transfer of data (e.g., EU Data Protection Law [12]). From a technical perspective, multiple geographical locations maybe required for reliability and disaster tolerance. As disaster tolerance requires replicating services in geographically diverse sites. As a result of these factors, today's cloud infrastructure providers typically operate a few very large data centres. These data centres are usually located in a small number of carefully selected geographical locations.

Connectivity between data centres owned by a single provider is usually implemented by leased virtual networks providing static, but guaranteed quality of service to the IaaS owner. Connectivity between an IaaS user and the data centre is generally handled by the Internet. Hence the user's network experience is based on access to a shared medium, which is **not** under the control of cloud providers. It is possible to dynamically scale the infrastructure implemented by an IaaS provider at low cost. However, it is hard to provide low cost dynamically scalable connectivity to that infrastructure - as networking is relatively speaking **more** expensive than storage or computing. To provide greater security, IaaS providers have recently added VPN tunnelling connectivity for their customers using protocols such as IPSec (e.g., as used in Amazon’s Virtual Private Cloud [17]). This enables the creation of an Information Technology (IT) infrastructure in the cloud that is connected to the site network of an enterprise, thus enabling the enterprise to use their own IP address space and network services across both their site network and the cloud extension of this network. Network limitations such as bandwidth, jitter, and latency offered by their Internet service provider and the lack of support for dynamic provisioning of network capacity are some of the issues that need to be addressed.

Applications such as large scale simulations, graphics rendering, on-line web services, and hosted IT systems are currently deployed in cloud environments since these are well suited applications for this architecture. Where sensitivity to network performance is an issue, such as content delivery [18], it is still necessary for the service provider to own the infrastructure or to enter into a long term contractual agreement with the infrastructure provider. The network components and topology of these services are generally not very dynamic.

2.2 Virtualization technology supporting Cloud Networking

Network virtualization brings a missing piece to the cloud computing puzzle. Virtual networks are not new in themselves. In [19], Mosharaf, Chowdhury, and Raouf Boutaba survey the technologies used at various layers to provide virtual networks. A number of network virtualization architectures and frameworks have been proposed in the literature, including VINI [20], CABO [21], 4WARD VNet [22], and FEDERICA [23] to offer customised virtual networks with end-to-end control by the cloud provider.

The possibility to specify and instantiate networks on demand and within a useful time period is one of the great advantages of network virtualization. Virtual networks can be created to meet different requirements, such as providing a specified bandwidth, meeting a specific end-to-end delay bound, offering a particular set of security features, and must support a selected set of protocols. Network virtualization introduces other advantages, such as the ability to reconfigure the network in real-time without losing connectivity, to change the physical path, or even to move one or more virtual nodes from one place to another [24]. Cloud networking enables network virtualization beyond the data centre bringing two new aspects to conventional cloud computing: the ability to connect the user to services in the cloud and the ability to interconnect services that are geographically distributed across different cloud infrastructures. Cloud networking users can specify their required virtual networking and computing infrastructure and the desired networking properties that they require in order to access these resources. Cloud networking users can specify how their infrastructure should be distributed in space and how it should be interconnected. These users should be able to do this dynamically, on-demand, and through a single control interface in a similar fashion to which they can manage virtual machines in a cloud.

The development of cloud computing has also encouraged the automation of services. Applications which run in a cloud environment can be programmed to monitor their own resource usage and depending on the load (or pattern of the load) they can dynamically scale themselves without the intervention of a human operator. Similarly, IaaS management systems intelligently optimise the use of available physical resources by automatically deploying and migrating virtual machines. Introduction of virtual networks to the same control plane (i.e., the same management system) will enable both the providers and users to make optimisation decisions based on network conditions as well as the factors that they currently consider. Moreover, as the variety of applications running in the cloud increases new requirements are introduced due to these applications. In most cases it may be better to deploy processing and storage operations distributed across a network, preferably bring the critical resources, closer to the user, rather than utilizing a centralised processing and storage facility. Network impairments, such as latency and jitter may hinder the real-time execution of certain cloud applications in a centralised infrastructure. Depending on the usage patterns, servers within a certain network delay bound of the user must be utilized, thus limiting these servers to be those located within a certain geographical region. As a result having a geographically distributed cloud enables greater control over the user’s experience.

Virtual desktop services and content distribution services are examples of this class of virtual applications that may need to be located near to the user in a network delay sense. Offering a wider range of trade-offs between costs and performance requirements requires a wider range of deployment options. To enable these new possibilities, it is important to understand the security requirements and to build appropriate mechanisms into the technologies we develop. Security is a major factor influencing the acceptance of cloud computing in real world deployments, especially when sensitive information will be processed and stored in the cloud. When establishing Service level agreements (SLAs), the critical areas of focus such as storage and computing in the cloud must built upon existing well defined IT security guidance [25]. From a cloud user's perspective security topics can be divided into application security, infrastructure and platform security, compliance, and governance [26]. The strength of a solution which addresses these topics can be evaluated based upon which security objectives are met, who is allowed to do what (authentication & authorisation), how are system elements and data protected (availability, confidentiality & integrity), how can the requirements of security policies be validated and checked (auditing), and how can the cloud provider prevent others from misusing the resources and prevent forbidden operations (misuse protection). On one hand, cloud networking adds another layer of security challenges to the existing cloud computing security issues, arising from additional networking capabilities. While on the other hand, cloud networking has the potential to control the existing cloud computing deployment models, thus solving other security problems which otherwise might negatively impact the acceptance of this technology.

2.3 CloNe Architecture

The CloNe high level architecture consists of four parts: a three layer model, a set of roles, a set of interfaces by which these roles interact, and a set of management functions in which these roles participate [1]. This model is a framework for portraying the virtual infrastructure relative to three different view-points namely roles, interfaces, and management functions. This section describes each of these parts.

Figure 1 illustrates the CloNe architecture [1]. An administrative domain is a collection of virtual or physical resources that is under the management of a single administrative authority. An infrastructure may span multiple administrative domains, but a virtual infrastructure exists within a single administrative domain. Three different administrative domains are shown in Figure 1. The resource, single-domain infrastructure, and cross-domain infrastructure are the three different views on which different roles, interfaces, and functions relate. The management of virtual resources and the construction of the three layer model are influenced by the way authority is distributed over administrative domains.

2.3.1 Resources

A virtual resource is an abstract representation of an element of the virtual infrastructure such as a logical volume on a block device or a virtual machine. A virtual resource is always located within the limits of a single administrative domain. The resource layer consists of the computing, storage, and network resources as virtual entities. Each of these different types of resources is generally managed by different sub-systems. Resources have identity properties and status. Resources may also have connections to other resources within a single administrative domain. A property is an externally determined attributed such as a networking address space for a sub-net or memory size for a virtual machine. A status is internally determined and it reflects the condition of a virtual resource: a life cycle stage or an error condition.

A virtual resource can be created dynamically, managed, and destroyed. The control actions typically are carried out by a subsystem such as a storage device manager, a storage array control system, or the management interface of a virtual machine hypervisor. Virtual resources within one administrative domain are often connected to other virtual resources of the same administrative domain, thus a virtual machine may be connected to a storage device, and/or virtual network. These connections establish relationships. The condition of this relationship can be validated if the virtual resource can be correctly established with its own rights or if the relationship depends on the properties and status of the related virtual resources. Although a virtual resource will be managed by the management interface of a single administrative domain, it may have connections with virtual resources in other administrative domains.

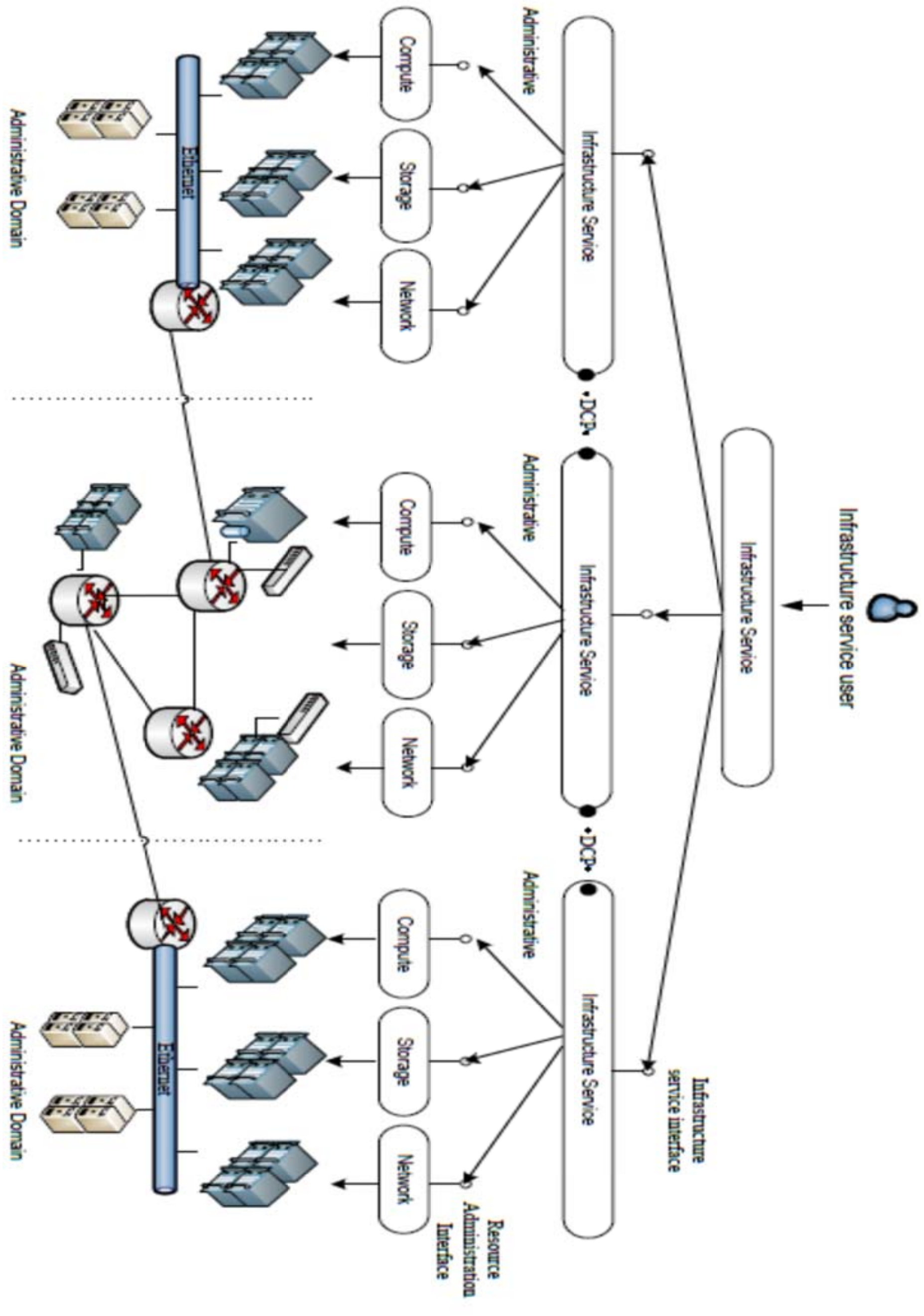


Figure 1: High-level CloNe architecture

2.3.3 Single-Domain Infrastructure

A single-domain infrastructure consists of a group of virtual resources managed by a management interface, collectively within a single administrative domain. The relationships between these virtual resources determine the topology of the infrastructure and constrain their combined management behaviour. Within a single administrative domain the administrative authority has full knowledge about all the available virtual resources and other virtualization capabilities at any time.

A single administrative domain infrastructure can dynamically be created, updated, managed, and destroyed. Mapping between the single-domain infrastructure and the underlying equipment can be determined at this layer by the administrative authority. This mapping can take into account the group allocation and can be used to provide optimal placement of virtual resources relative to each other. For example for optimal network performance a Virtual Machine (VM) might need to be running in a particular location. Moreover technology selections can be made at this layer. A VM could be executed on a choice of the type of servers these could be characterized by having different processor chips or different memory sizes enabling different performance trade-offs; a disk volume could be placed on a local storage device or be located in a network attached storage service; a network connection could be mapped to an open shared network or an isolated VPN tunnel. Optimal placement and the choice of the most preferred technology will depend for the most part on private policies of the administrative authority for each domain.

2.3.4 Cross-Domain Infrastructure

A cross-domain infrastructure consists of a number of virtual resources managed individually by management interfaces collectively across multiple administrative domains. A cross-domain infrastructure can be partitioned into multiple single-domain infrastructures. A single domain infrastructure may contain resources that have connections with virtual resources in other single administrative-domain infrastructures, thus interconnecting the virtual infrastructures and determining the topology of the cross-domain infrastructure. A cross-domain infrastructure is managed by multiple administrative authorities. In contrast to a single-domain infrastructure, the state of underlying equipment and virtualization capabilities is unlikely to be completed shared beyond a single administrative domain's boundaries. To allow cross-domain management there needs to be interfaces via which resource virtualization can be negotiated. Via these interfaces, the authorities of the different administrative domains may exchange information that they are willing to share about resources in their domain in order to facilitate cross-domain virtualization optimization. A cross-domain infrastructure can be created, updated, managed, and destroyed. Partitioning of the virtual infrastructure into administrative domains can be performed at this level based on the capabilities of the administrative domains and their interconnection. Properties and connections of the virtual resources and properties of the virtual infrastructure as a collection will also influence this partitioning. The resulting hierarchical structure of interfaces is shown in Figure 2.

2.3.5 Roles

In the SAIL architecture we broadly divide each user entity into roles. The three roles that are assumed here are:

1. **An Administrator** has the administrative authority over underlying the virtual or physical resources. This administrator can use management systems to configure and manage all of these resources within their administrative domain.
2. **An Infrastructure Service User** accesses the infrastructure services in order to create, examine, modify, and destroy virtual resources.

3. **An Infrastructure Service Provider** provides the infrastructure service that may be used by an Administrator to give access to an infrastructure service user so that this user can in turn create, examine, modify, and destroy virtual resources.

2.3.6 Interfaces

Three types of interfaces are considered for these different roles. These interfaces are: (1) Resource administration interfaces, (2) Distributed control plane, and (3) Infrastructure service interface.

The resource administration interfaces implement the management functions used by the administrator to create, manage, and destroy virtual resources within their own administrative domain. These interfaces are realized by the management interfaces of some virtualization technology; hence usually they are implementation specific. These interfaces provide information about the underlying infrastructure including the technologies used, so that the administrator can decide how these resources should be managed efficiently and what information needs to be passed through these interfaces. Each interface can take specific configuration details from an administrator (concerning specific compute, storage, or network resources) and configure these resources according to the infrastructure service user's requirements. These resource administration interfaces are further divided into computing Resource Interface, Storage Resource Interface, and Network Resource Interface.

1. **Compute Resource Interface**: This interface can be used to perform the following operations on VMs: Create/Start/Delete/Suspend/Stop, selects the Software OS and execution environment will be executed on a given VM.
2. **Storage Resource Interface**: Different types of storage (ranging from traditional servers, Storage area network (SAN) or storage in the network nodes) is managed via this interface.
3. **Network Resource Interface**: This management interface will have an overall view of the underlying network. An isolated path through the network can be allocated for users with special needs for specific parameters such as jitter and bandwidth. This network resource interface has a centralised knowledge about the whole network infrastructure under control of an administrator.

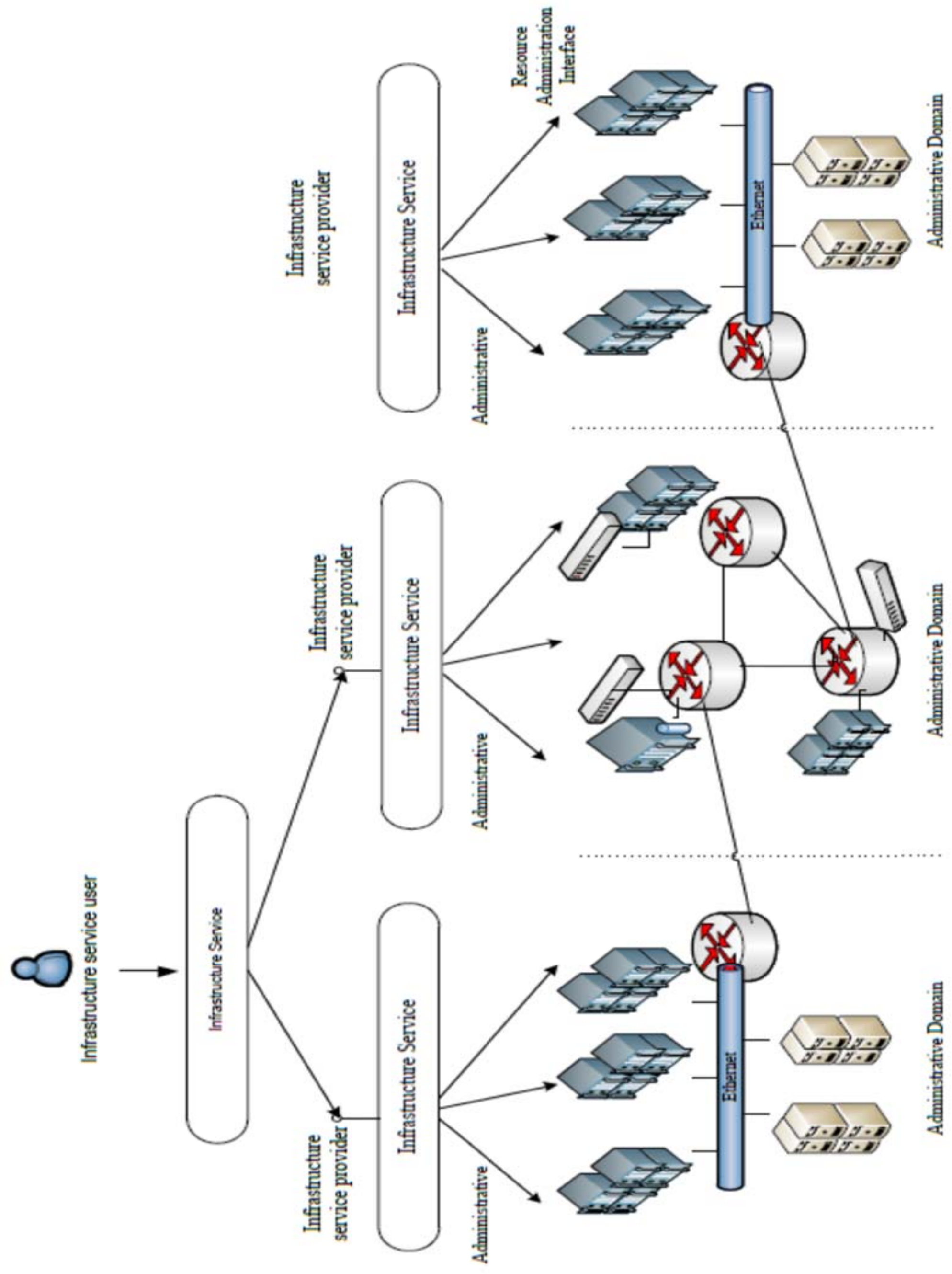


Figure 2: Hierarchical interaction between the CloNe interfaces

2.3.7 Distributed Control Plane

The Distributed Control Plane (DCP) describes the collection of protocols, interfaces, and control operations that enable more than one infrastructure service providers to interact and exchange cross administrative domain information. As a result the DCP is located at the cross-domain infrastructure layer. For example, if two neighbouring administrative domain want to implement a network link between their domains, then both of them need to know how they are connected to each other (network edges, technologies, negotiating protocols, and other parameters). Similarly, two infrastructure services may need to interact to coordinate a management operation [1]. Communication between domains on via DCP need not be synchronous. The configuration of a domain if needed can be passed to others depending on the specific relationship with these other domains and the technologies used. The specific protocols used will depend on the relationship between domains and technology used in the domains. However, generic message passing can be employed to communicate the common parameters.

2.3.8 Infrastructure service

The infrastructure service is one of the most fundamental parts of the CloNe architecture. The infrastructure service provides a set of interfaces that enable the creation, monitoring, and management of virtual infrastructures provided by the infrastructure service provider role and accessed by the infrastructure service user role.

In SAIL [1] we assume that a user request to the infrastructure service will be made using a high level description language. The objective of using such a language is that the specified high level goal can be used to form the system service level agreements. These SLAs will be broken down automatically by the system to realize low level control actions.

2.4 Networking Technologies in CloNe

The unification of cloud computing and virtualized network provisioning introduces the need for new management functions. Unfortunately, the generic lack of predictability of the cloud, coupled with ever increasing network complexity has decreased the overall dependability levels of cloud systems. Networking, specifically for the cloud, has its own set of problems. These include equipment heterogeneity, resource management, varied degrees of access to underlying network information, fast and timely reconfiguration (with most of these problems related to virtualization), and/or interoperability challenges.

The focus of a cloud network is to ensure more agile network models, as compared to previous static networks, and to deploy them at a lower cost. At best, a network custom made for the cloud should support the cloud infrastructure’s required characteristics, for example it should be dynamic, allowing on demand creation and migration of IT resources, thereby improving the overall quality and performance (i.e., offering higher throughput, reduced jitter, and low latency). Both local area networks (LANs) and wide area networks (WANs) should be considered when building the underlying network for cloud systems. The design of networks for cloud systems has increased complexity due to the use of virtual servers, since virtual servers allow dynamic creation, configuration, and deletion of virtual machines, which otherwise would be more a static process. LANs customized for the cloud will probably exploit Ethernet as the single data centre switching fabric, eventually displacing technologies such as fibre channel that are used today for storage networking. To improve server-to-server communications, the conventional three-tier data centre networks can be re-engineered to create two-tiers: an access layer and the core/aggregation layer. This two tier model can improve network performance (such as reducing the total number of hops between servers, or by efficiently prioritizing different types of traffic due to its centric nature.

A typical two-tier network supports the server virtualization topologies, by using technologies like VLANs and VxLANs [16]. These virtual LANs may be extended throughout the data centre, to support dynamic VM migration at layer 2. Networking requirements for virtual servers built on top of multi-core physical servers can exceed the capacity of Gigabit Ethernet (GbE) and multi-GbE aggregated links. Multiple virtual servers that maybe running on a single processor, hence with the increase in total number of cores in the underlying physical server the result is a proportional increase in the I/O requirements. A datacentre LAN with redundant links can utilize the parallel links between the servers to the access layer and also the multiple parallel links from the access layer to the core layer. Loops can be eliminated in the logical topology using technologies such as switch virtualization and multi-chassis link aggregation groups (LAG), which enable utilization of all available resources.

In switch virtualization, two or more switches are made to appear as a single logical switch, with a single control plane, to other elements of the network. Virtual switch links (VSLs) or virtual switch interconnects (VSI) are needed by the elements of the virtual switch to communicate. Multi-chassis link aggregation group (LAG) technology allows the links of the aggregation linkz to span the multiple physical switches that together comprise a single virtual switch. Combining switch virtualization and multi-chassis LAG we can create a logical loop free topology and utilize all available resources at the same time[*]. Loops are avoided using these technologies, because from a logical perspective the two switches appear as a single virtual switch. Moreover, traffic to and from these servers are load balanced across the two links participating in the multi-chassis LAG making all ports active ports. These technologies can be further integrated with other protocols such as the spanning tree protocol (STP). Switch virtualization aggregates a smaller number of switches to be a group. More since all the switches maintain the same state, less effort is required to maintain this state. Transparent Interconnection of Lots of Links (TRILL) [17] is another approach to designing layer 2 shortest-path first (SPF) forwarding protocol for Ethernets. With TRILL it is possible to achieve load-balanced, active-active link redundancy which can be combined with switch virtualization and VSL/VSI interconnects, thus producing the next generation of data centre networks.

A CloNe flash network slice (FNS) represents an abstraction of the basic network resources that are part of the CloNe architecture. FNS can be achieved using various networking technologies, but the focus of FNS is to provide: (1) efficient provisioning, (2) traffic support, and (3) ease of management. A number of different networking technologies can be used to make it possible to quickly and dynamically set up, modify, and tear down a FNS. This can be used to allow a wide range of different traffic classes to be supported in an efficient way; while simplify the network as seen by the user, since we are able to abstract away unnecessary details. With FNS we also provide a sufficiently rich set of operations to enable the user to control the FNS according to his/her needs.

Virtual private networks (VPNs) are the building block of today's enterprise networks and this is unlikely to change significantly in the near future. Use of VPNs has been very successful in the enterprise market, as this shifts the task of operating the complex networks that connect different sites to the service providers. By using VPN tunnelling with encryption technology (for example, using IPSec) a minimum default level of security is provided by the network. This tunnelling allows private addressing to be extended from the enterprise to the data centre enabling the aggregated hybrid usage of both customer premises resources and cloud resources. This also means that migration of resources to and from the cloud becomes easy. In the case of VPNs operated by the service provider higher levels of security and reliability can be part of the SLA.

[*] This is not possible when using the spanning tree protocol (STP) and the virtual router redundancy protocol (VRRP) since both prevents the available forwarding resources in a redundant network design from being simultaneously utilized.

Since the encryption/ decryption keys are maintained by the service provider this type of VPN is vulnerable to inappropriate data access by the service provider. Unfortunately, VPNs are not cloud ready [18], since the use of VPNs leads to a rigid model, while to maximally exploit the cloud the resources must be customer configurable. Elasticity of resources (e.g., adding new VPN sites), resource mobility, on-demand reconfiguration (changing bandwidth capacity), have rarely been the requirements of service provider's VPNs, as their VPNs are expected to be a relatively stable service offering, i.e., with relatively few configuration changes. When an internet service provider (ISP) provides VPNs, the customer's perspective of a VPN is almost a black box, since the VPN resources are mostly controlled by the VPN service provider. In order to make such VPNs cloud ready some layer of abstraction is necessary so that customers can control the functionality of the VPN to a certain degree.

2.5 FNS with Flow based networking

OpenFlow is an open standard that enables flow level control by separating the control and forwarding plane. OpenFlow switches consist of flow tables with associated actions in the flow table entry. An OpenFlow controller, which is open source and programmable, computes these flow tables and injects them to OpenFlow switches. OpenFlow requires only a relatively simple physical switching infrastructure, which facilitates hardware virtualization. OpenFlow enables online creation, modification, and migration of network controllers. As a result, network resources can be more dynamically and flexibly created and modified than in the traditional networking approach. OpenFlow rules, which are simple rules such as access control lists (ACLs), can be injected either on a permanent basis or on demand as temporary entities. Permanent injection has less scalability, but it is faster to process the rule. An intermediate virtualization controller, such as a FlowVisor, which acts like a transparent proxy server between the switches and the relevant controllers, can slice an OpenFlow switch into several virtual switches making it possible for OpenFlow controllers, to control virtual OpenFlow switches. Dynamic creation and modification of FNS can be achieved by changing the FlowVisor configuration on-demand. Therefore, OpenFlow can meet the overall requirements to enable virtualised network resource provisioning. However, the overall network requirements still do not yet address the security flaws covered in section 2.6. Hence the deployment of a multi-level tightly integrated security architecture to strengthen the overall CloNe architecture is still necessary.

2.6 Security flaws and goals of CloNe

This section covers a subset of the overall security flaws that plague existing cloud network architectures. This section also describes the security goals which need to be achieved in order to attain an acceptable level of security. It is imperative to identify a comprehensive list of security challenges which affect the cloud networking ecosystem. Schoo et al. [44] provides a suitable source for these, by describing challenges pertaining to information security in clouds, communication security, and virtualization environment threats. In order to narrow down the list of plausible security challenges, it is important to understand the network resource provisioning model defined by CloNe. Moreover, CloNe has developed a concept known as the Flash Network Slice (FNS). The FNS is a virtual network resource which allows dynamic network resource provisioning capabilities in an operator controlled network environment, such as WANs, while making use of distributed processing. A FNS is a resource which provides a network service. A FNS can have multiple access-points and implements forwarding between those access points. A FNS can be linked to other resources through connections. A FNS can be provisioned inside a single administrative domain (i.e., a single operator controlled environment). This FNS should have measurable and acceptable QoS and setup times. Based upon the networking requirements of the FNS, a list of security goals and requirements has been generated (see section 7.1 of [45]).

The CloNe architecture requires meeting security goals similar to any other computing infrastructure, namely availability, integrity, confidentiality, authenticity, non-repudiation, and privacy. These security goals, with respect to the CloNe infrastructure, are considered to be concrete security properties that are desired to be provisioned for the overall CloNe architecture. On the other hand, these security requirements are also considered to be high-level security specifications which would be realized by implementing a chosen set of security goals. For example, isolation, a security requirement, requires confidentiality, privacy, and integrity (each security goals) for its implementation.

In order to better understand the overall security requirements, it is important to describe two distinct application scenarios of CloNe. The first is termed "Dynamic enterprise", as it entails provisioning of IT solutions from the cloud network ecosystem to the enterprise market. The second scenario is termed "Distributed cloud: Elastic video delivery", as it allows a service provider to offer real time video via a cloud to consumers. The first scenario requires an enterprise centric cloud networking solution which ensures full resource isolation between tenants (both within the WAN and within the data centres), dynamic assignment of network resources in the datacentre and WAN, dynamic scaling of virtual resources (for example computing and storage), and dynamic provisioning and scaling of network resources such as bandwidth. The "Dynamic enterprise" use case is applicable when the infrastructure of an enterprise is partially/wholly shifted into the cloud. In contrast the second scenario provides real time video via a cloud to consumers, and requires cloud network capabilities to provision & scale with the allocation of distributed virtual resources spread over an operator's network, distributed load balancing, and optimal placement of content servers in the distributed cloud according to relevant measures of optimality.

The security requirements include high-level specifications including information security, virtualization management, isolation, misuse protection, DoS protection, and identity management. Among these identity management is an extremely important requirement, especially due to the strong focus of CloNe towards preventing malicious entities communicating with legitimate entries as part of CloNe service offerings. For example, a service provider masquerading as Deutsche Telekom could accept a service request from an important customer, and then misuse the data provided to it as part of the virtual resource provisioning request. Identity management also supports other security functions that are being planned as a part of the overall CloNe security architecture, namely the access control mechanism and the overall security goal translation function. There is a need for a well-integrated identity management framework in order to develop and manage identities and to realize access control policies relevant to the different tenants. Moreover, the different entities involved in the architecture must be authenticated, and their access to information and services should be verified against their permissible usage profiles and access policies. This authentication and secure data transmission will utilize a key management algorithm, customized for CloNe. Therefore, a suitable new key management algorithm was developed during this thesis project.

A centralized security goal translation function was also developed as a part of this thesis project. This security goal translation function accepts security requirements from all the participating entities in the CloNe infrastructure, then translate them into resource configurations which will be deployed by the resource administrator on the underlying set of resources. The input requirements detail the varied security requirements of the different partners, and the translator should ensure that the overall infrastructure can achieve the security levels requested by the different entities. The security goal translation function incorporates an intrusion detection system, details of which will be covered later in Chapter 8. This intrusion detection system will provide an initial auditing mechanism, in order to help detect intrusions and also to provide scope for learning by the security goal translation function. This learning can help the translator to better equip itself against future attacks. The final components of this thesis give details of this central security goal translation function, which is expected to utilize two central supporting security functions, specifically the key management algorithm (as a part of an access control mechanism) and an intrusion detection mechanism. Both of these security functions will be customized by considering the overall CloNe security requirements and the two use case scenarios of CloNe, and the CloNe architecture specifications.

A description of the security architecture for CloNe is given below. In the next chapter a comparison with other security architectures will be given in section 3.1. In sections 3.2 and 3.3 (respectively) a survey of the access control mechanisms and key management algorithms is provided. A survey of intrusion detection systems is given in section 3.4. This background material will complete the description of the state of the art for the proposed security functions, specifically the key management function and intrusion detection system, which are to be developed as a part of this thesis project.

2.7 CloNe security architecture

This subsection describes the initial CloNe security architecture which was defined as a part of the SAIL project. The existing security architecture has the central aim to allocate (and migrate) virtual resources requested by the infrastructure service user amongst the different service providers [46]. However, this allocation should also respect the security requirements specified by the infrastructure service user and the infrastructure service providers. Moreover, there are additional requirements that need to be adhered to. For example, legal regulations and organizational requirements which need to be adhered to with respect to the service being offered, organizations to whom the cloud service is being provisioned, and the geographic location of the physical resources of the service provider.

The security architecture needs to ensure that the entire process of resource allocation with respect to the specified security requirements is executed automatically and with low overhead. Moreover, the entire process needs to be efficient and accurate (i.e., having a low enough failure rate that it is acceptable to all participating entities in the CloNe infrastructure). Meeting these requirements will be a major improvement to the state of the art, which currently requires that the infrastructure service user manually compare the prices and security levels of the infrastructure service providers, and it also requires the infrastructure service provider to manually execute the migration and placement process.

The security goal translation process is initiated by an infrastructure service user by describing the security goals which need to be realized by the underlying set of resources. Once the security goals have been defined in the form of a security policy (for example, the goals adhere to a set security standards which conforms to their overall organizational and operational policies), then the goal translation function shall translate these goals into security parameters. CloNe has demarcated a reduced list of security parameters for prototyping purposes. The highest priority parameters for CloNe are access control specific parameters, identity management related parameters, and geographic location specific parameters. Clearly, as noted earlier, security requirements are specified not only by the infrastructure service user, but by the infrastructure service provider as well. A classic case could be that the user is neutral to the geographic location of the resource set as long as their requisite price point is met. However, the infrastructure service provider may have specific operational policies for different services, for example in order to implement elastic video distribution to clients in Europe; the infrastructure service provider might not want to place the physical resources in Asia. Figure 3 illustrates how security parameters are generated by combining the inputs from both the infrastructure service user and the infrastructure service provider. The service provider will realize these parameters by combinations of security services and mechanisms as shown in Figure 4.

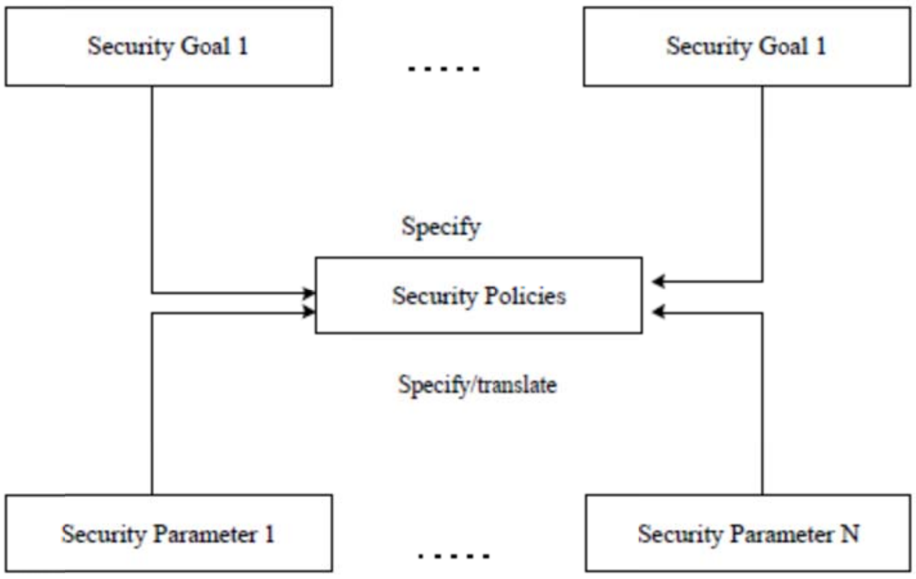


Figure 3: Generation of security parameters of users

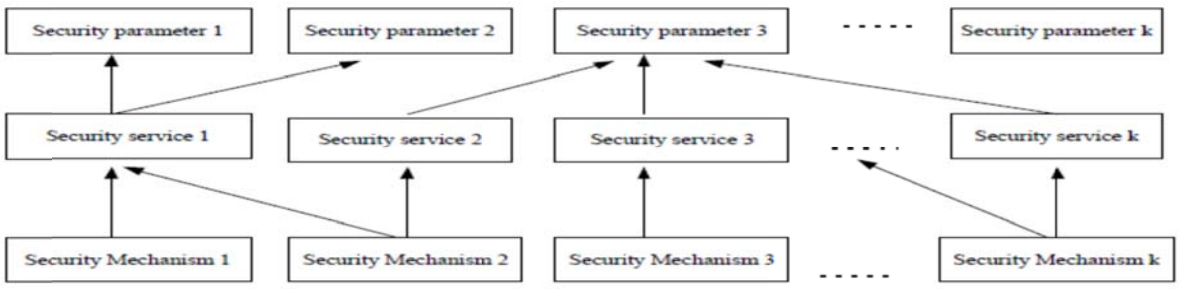


Figure 4: Service provider

Next logical step would be for the security goal translation function to translate these security goals into resource constraints, which need to be implemented by the underlying resource set, as show in Figure 5. Security goals, which were defined in the form of security policies, have been used to specify the security parameters. These security parameters will in turn be translated into resource constraints, which determine the resource configurations deployed on the underlying resource set. This process was further extended during this thesis project. The resource constraints will invoke security services deployed by each infrastructure service provider. For example, each infrastructure service provider is expected to implement their own access control policy models, identity management solutions, and auditing mechanisms. These services will then implement the fine grained security mechanisms. Security mechanisms that can be used to implement an identity management service include authentication, authorization, and organizational policy compliance mechanisms.

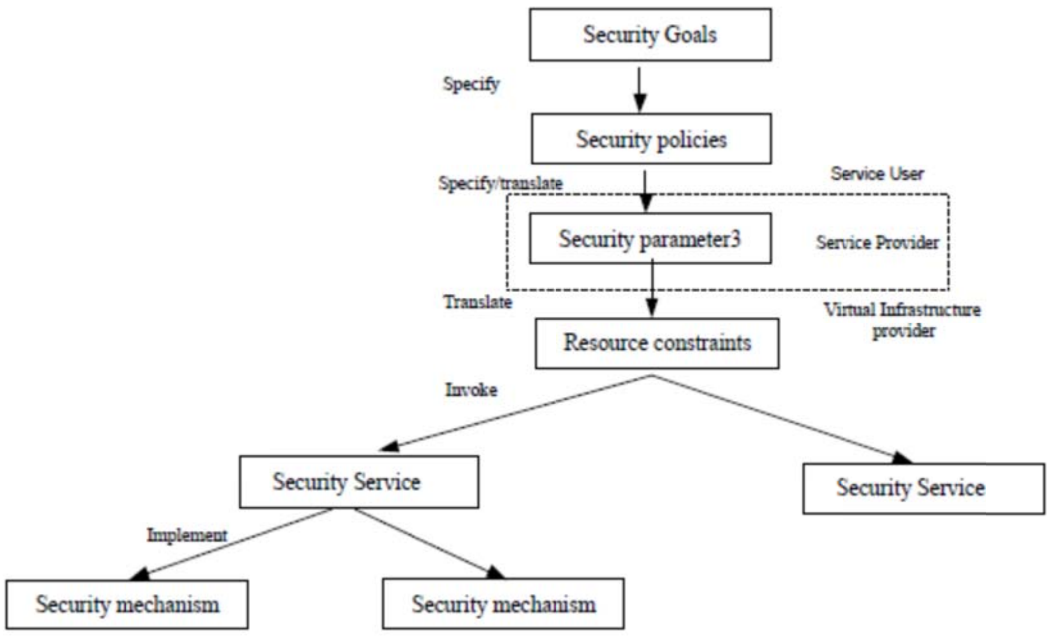


Figure 5: Abstract representation of security goal translation [46]

Clearly, even though the CloNe security architecture is still a nascent stage, it offers a stable backbone infrastructure to the overall CloNe architecture. Its salient features include access control, on-demand secure virtual storage provisioning, on-demand virtual compute provisioning, on-demand virtual network provisioning, secure multi-domain communication, secure VM migration between domains, multi-pronged and comprehensive identity management solution, and multi-objective security goal translation (this security goal translation considers the objectives specified by *all* the participating entities).

3 Background

This section reviews the state of the art regarding security architectures for cloud provisioning models, access control mechanisms, key management methodologies, and intrusion detection mechanisms. Where applicable, weaknesses of the existing methods are stated, and potential improvements considered for future development are mentioned.

3.1 Related works concerning cloud security architectures

Currently, there are multiple (security) architectures and toolkits (in addition to CloNe) that provide and strengthen the cloud delivery models. The most important of these are the Open Security Architecture, Amazon EC2 Security Architecture, and the GRC Stack developed by the Cloud Security Alliance. This section describes each of these and then offers a comparison between these architectures and the security architecture of CloNe described in Chapter 2. The comparison is based on well accepted parameters in the cloud service provisioning ecosystem, and aims to reflect the overall dependability and performance characteristics of the underlying infrastructure. These parameters include access control, on-demand secure virtual storage provisioning, on-demand virtual compute provisioning, on-demand virtual network provisioning, secure multi-domain communication, secure VM migration between domains, identity management solution, support for hybrid cloud computing, multi-objective security goal translation, on-demand secure network scalability, and multi-level security. Although the CloNe security architecture is only at the conceptual stage, this thesis project will elaborate the overall architecture development of the CloNe security architecture.

The Open Security Architecture [47] has been released by the OSA (a not for profit organization). Its main purpose is to release best practices, security patterns, and architectures to help strengthen widely used (security) systems. The supporting security services are well integrated into the overall OSA architecture. Their architecture supports both on-demand secure virtual storage and compute provisioning. However, due to an absence of a virtual network resource provisioning ability in their architecture, their architecture cannot (itself) securely provision network resources. Similar to other architectures in the cloud ecosystem, their architecture supports the introduction of identity management solutions, although it is not as fine grained or as detailed as CloNe's proposed security architecture. Both secure multi-domain communication and secure VM migration between domains are omitted from their architecture, which from the SAIL perspective is a major flaw in their architecture. Systems will need to integrate interaction between different administrative domains securely within their existing delivery models, especially if the user's resources are stored at one operator, but the desired computation requires a resource set which cannot be provisioned by the user's current infrastructure provider due to unavailability and/or economic/organizational factors. OSA supports hybrid cloud computing, which allows users to pick and choose their final delivery models. Moreover, OSA supports multi-level security, which provides a second (and sometimes third) line of defence. To conclude, OSA has no support for multi-objective security goal translation or on-demand secure network scalability, making it unfit for a multi-domain and multi-operator scenario.

Amazon has introduced their elastic compute cloud, or EC2, which is Amazon's web service which provides resizable compute capacity in the cloud [48]. Moreover, the GRC stack by the Cloud Security Alliance provides an exhaustive toolkit to instrument and assess both private and public clouds against industry established best practices, standards, and critical compliance requirements [49]. Unfortunately, both the EC2 and GRC stack service models contain the same shortcomings as the Open Security Architecture, discussed above. This renders all three service models unsuitable for a multi-domain, multi-level service provisioning model with multiple stakeholders and participating entities.

These three cloud security architectures exemplify the current situation of cloud service delivery models, where vendor lock-in is commonplace, and inter-operator communication, let alone secure interaction and trust management, is extremely rare. On the other hand, in the case of EC2, there are compatible architectures that would allow a user to combine the user's own cloud with EC2's cloud. However, it (multi-domain provisioning) is still not a functionality which is integrated into the original resource model provisioned by Amazon and requires additional tweaking by the customer and/or intermediate service providers.

In comparison, CloNe (and especially its security architecture) encourages and supports secure interaction and trust management between different cloud service providers. The CloNe security architecture aims to have a well-defined and multi-grained access control policy function, which can accept access control policies from the different entities participating in the architecture and deploys these policies on the underlying resource set. This architecture supports all three on-demand secure virtualized service provisioning models (on-demand secure virtual storage provisioning, on-demand virtual compute provisioning, and on-demand virtual network provisioning), thus raising the overall dependability levels of the offered service by invoking the network guarantees established the SLAs of the provisioned services. As stated earlier, the architecture encourages inter-operator communication and multi-operator service delivery models by supporting both secure multi-domain communication and secure VM migration between domains. Additionally, the architecture supports hybrid computing which enables the user to choose her preferred delivery model based on her key performance indicators. The CloNe security architecture is bolstered by its support for on-demand secure network scalability, but is hampered due to the absence of multi-level security. A comparison of the different architectures is shown in Table 1.

Table 1: Comparison of different architectures

	CloNe Security Architecture	Open Security Architecture	Amazon EC2	CSA GRC Stack
Access control	Yes	Yes	Yes	Yes
On-demand secure virtual storage provisioning	Yes	Yes	Yes	Yes
On-demand virtual compute provisioning	Yes	Yes	Yes	Yes
On-demand virtual network provisioning	Yes	-	-	-
Secure multi-domain communication	Yes	-	-	-
Secure VM migration between domains	Yes	-	-	-
Identity management solution	Yes	Yes	Yes	Yes
Support for hybrid cloud computing	Yes	Yes	Yes	Yes
Multi-objective security goal translation	Yes	-	-	-
On-demand secure network scalability	Yes	-	-	-
Multi-level security	-	Yes	Yes	Yes

3.2 Related work on access control mechanisms

This section compares the existing access control mechanisms, including mechanisms that operate at the virtual resource level. At the virtual resource level, the best possible options for access control mechanisms will be described. Additionally, the access control mechanisms that need to be implemented at the physical resource level will be covered.

Damiani et al. [71] proposed the notion of a fine grained access control model for XML documents. In contrast, many researchers [72,83] have focused on controlling access to XML documents by providing new definitions and enforcement of access restrictions directly on the structure and content of the XML documents. However, all these efforts have focused mainly on protecting XML documents for web services, rather than providing access control for web databases. Xianzhi Huang et al. [73] have described the access control policies for XML using regular path expressions (such as Xpath) for specifying the object to which the access control policies apply. Even though, this work is more suitable for an enterprise computing environment, this method places an additional load on the query engines when accessing XML documents. To reduce this burden, these authors introduced a static analysis model for their access control subsystem. However, dynamic analysis is necessary to provide security in a web services and web database environment.

The Discretionary Access Control (DAC) [50] model utilizes the Harrisson, Ruzzo, and Ullman (HRU) model which defines access rights of the system entities, by using subjects, objects, and actions. The subjects are the actors who can execute actions over other actors (objects). Thus, subjects are active, while the objects could be both active and passive entities. The CloNe security architecture allows the infrastructure service users and infrastructure service providers to specify security requirements, which define the access control policies for every user/service. Therefore, a security requirement specified by the infrastructure service provider would define an access control policy (action) for an infrastructure service user (subject), with respect to a cloud infrastructure service (object). Each security requirement is described in terms of a set of actions A(s,o) which can be performed by the subject s on the object o.

Disadvantages of DAC include a lack of scalability of the system with respect to additions of new participating entities. This is due to the need to update all of the security policies for every additional participating entity. Moreover, advanced security policy support through the use of prohibitions, obligations, or recommendations is absent. Only the top-level administrator has the permissions necessary to update the overall security policy set. An alternative approach is role-based access control (RBAC) [51]. RBAC has a major advantage over DAC, due to the introduction of roles. Instead of assigning security policies for each and every infrastructure service user and infrastructure service provider separately, these access control rules can now be associated with roles. Additional features in RBAC include the introduction of a session and role hierarchy. A session is a period of time, during which each user can assume a subset of its complete set of roles. For example, during VoIP service delivery, an end user is only required to fulfil her VoIP customer roles, and need not perform any activities pertaining to her "mobile database consumer" or "elastic video consumer" roles. On the other hand, a role hierarchy arranges the possible roles in a hierarchical schema, allowing users to inherit roles. The RBAC model also supports constraints, which are used to specify special cases. For example, user A (low category consumer) could be prohibited from utilizing a mobile database and elastic video service simultaneously.

Feng He and Jia-Jin Lee [74] have presented a web services security technology for implementing RBAC components for a secure web services environment. Their work focuses on utilizing RBAC to protect e-learning applications based on web services. However, no complete access control model architecture for web services has yet been proposed by researchers that include rules, spatial constraints, and temporal constraints that can be intelligently handled by agents playing multiple roles.

There are many papers in the literature that deal with access control and data security. Ravi Sandhu et al. [75] presented a RBAC model based on users, roles, and operations. Recent interest in RBAC has been motivated by the use of roles at the application level to control access to application data and web data. Ninghui Li et al. [76] proposed an analysis technique to maintain desirable security administrator privileges; more specifically the authors defined a family of security analysis problems in the RBAC model. Barker [77] introduced a generalized RBAC model called the Action Status Access Control (ASAC) model based on autonomous changing of access control policies in response to events that involve agent action and exploiting a notion of status. Moreover, Barker has described the implementation of the ASAC model and performance measures. Fenghua Li et al. [78] compared the ASAC model with the other existing access control models and concluded that this model is best for open and distributed systems. Bertino et al. [79] proposed a Temporal-RBAC (TRBAC) model which is a temporal extension of RBAC. The main feature of this model is that it supports periodic enabling /disabling of roles and actions expressed by role triggers.

James Joshi et al. [80] proposed a generalized TRBAC (GTRBAC) model and specified various temporal constraints to use in user-role and role-permission assignment. Moreover, GTRBAC allows specification of a comprehensive set of temporal constraints, particularly constraints on role enabling and activation and various temporal restrictions on user role. They also presented time-based semantics of hierarchies and separation of duty (SoD) constraints. Here a notion of safeness has been introduced to generate a safe execution model for a GTRBAC system. Chen and Crampton [81] constructed a number of spatio-temporal role-based models based on RBAC [84]. Chen and Crampton also introduced a graph-based formalism to explain the semantics of RBAC, and used this as a basis for defining the semantics of spatio-temporal models. They also examined the difficulties that arise when enabling constraints are placed on roles in the presence of role hierarchy.

Xiutao Cui et al. [82] presented Ex-RBAC, an extension of the RBAC model adding identity constraints and spatio-temporal constraints in a location aware mobile collaboration system. These authors have also proposed some assignment rules of privilege in their paper.

The disadvantages of RBAC include the inefficient role hierarchy, which usually conflicts with the organization hierarchy, and there is not a natural mapping between them. There is an ambiguity about the differences between a role and a group, which further complicates the process of specifying permissions when defining access control policies. Additionally, it is not possible to specify access control rules which are only valid in a specific context. Only the top-level administrator has the necessary permissions to update the overall security policy set. Finally, the RBAC model does not scale well when the policy should transcend multiple organizational boundaries.

As evident from the above two access control policy models, they each have their specific drawbacks, thus a model is needed which avoids most of these drawbacks. Fortunately, the OrBAC model [52] shares none of these drawbacks and appears to fulfil the security requirements expected by the CloNe security goal translation function from an access control policy model.

The following features of OrBAC, offer a clear list of its advantages: OrBAC includes the concept of an *organization* allowing grouping and structuring of subjects, based on their roles. Thus, the role acts as a means to structure the association between subjects and their respective organizations. The concept of an *organization* allows the access control policy model to successively process multiple security policies spanning multiple organizations.

A huge boost in ease of scalability of access control policies with increasing organization/subject/roles size is possible because all three entities can be depicted using a single ternary relationship, as opposed to only a binary relationship within the RBAC model.

Similar to the relationship between roles and subjects, a *view* entity is introduced to structure *objects* and their respective organizations. A *view* entails all the objects which share a common feature.

If a *subject* is a concrete entity, then *roles* are the corresponding abstract variants of the same. A similar relationship exists between (concrete) *objects* and *view*. *Roles* abstract *subjects* based on common functions, while *view* abstract objects based on common properties.

An action is the core entity in access control policies, and defines the set of plausible actions that can be performed by the subject(s) on the object(s). *Actions* are abstracted using *activitiy* based on common shared principles.

Access control policies in OrBAC support permissions, prohibitions, obligations, and recommendations. The last option, viz. recommendation is not possible in other models, and these recommendations are extremely helpful for a green field model for cloud networking. The access control policies could be *soft,* for example provided as *recommendations* to the infrastructure service provider by the service broker working between the end user and a service provider.

OrBAC allows context-specific permissions and/or permissions to be specified as a part of the access control policies to be implemented on the underlying resource set.

Additional advantages of OrBAC include the concept of inheritance hierarchies, which have a natural mapping to network resources. Inheritance hierarchies are possible for *roles, views,* and *activities*, as well as for *organizations*. These features allow us to take the abstract security policies specified by the user, and then map them onto the concrete access control policies which can be deployed on the underlying network resources.

Moreover, Cuppens and Miege have proposed an administration model, AdOrBAC which allows role assignments to users, permissions, and permission assignment to users. AdOrBAC enables finer control over the delegation and collaboration activities covered in the SAIL project. Successively OrBAC can be controlled using MotOrBAC, which is a GUI built on top of the OrBAC API. This can be easily integrated into the planned prototypes for the SAIL project, and ensures rapid prototyping cycles to evaluate the performance of OrBAC in different case scenarios.

An initial security analysis of the CloNe architecture, OrBAC model, and the CloNe use cases have led to the need for a backbone key management algorithm that would be used to bolster the overall identity management function planned as a part of the CloNe security architecture. Access control policies (authorization) can only be implemented successfully between authenticated parties, which are not possible without key management. The literature about access control gives us a background to understand the needs for key management in SAIL.

3.3 Related work for Key management

There are numerous papers on key management and key distribution in the literature, such as [28], [29], [30]. In many of the existing key management schemes, different group of users obtain a new distributed multicast key for every session update. Among the various approaches to key distribution, the Maximum Distance Separable (MDS) [31] method focuses on the use of error control coding techniques for distributing re-keying information. In MDS, session keys are obtained by the group members based on erasure decoding functions [32]. In this method the group centre (GC) constructs a non-systematic MDS code C over the Galois Field GF(q) and a secure one-way hash function H(.) whose co-domain is GF(q). The GC generates n message symbols by passing the code words to an erasure decoding function. The first message symbol is considered a session key, out of the n message symbols, and the group members are not provided this particular key by the GC. Instead, the group members are given the (n-1) message symbols and they compute a code word for each of them. Each of the group members uses this code word and the remaining (n-1) message symbols to compute the session key. The main limitation of this mechanism is that it increases both computation and storage complexity. The computational complexity is proportional to $lr+(n-1)m$ where lr is the size of the r bit random number used in the mechanism and m is the number of message symbols to be sent by the GC to group members. If $lr=m=l$, then the computation complexity is nl. The storage complexity is $[\log_2 L]+t$ bits for each member. Where, L is the number of levels of the tree. Hence the GC has to store $n\ ([\log_2 L]+t\ )$ bits.

The data embedding mechanism proposed by Trappe et al. in [33] is used to transmit a re-keying message by *embedding* the re-keying information into the multimedia data. In this mechanism, the computational complexity is O(log n). The storage complexity is directly proportional to O(n) for the server machine and O(log n) for group members. This technique is used to update and maintain keys in a secure multimedia multicast via a media dependent channel. The main limitation of this mechanism is that a new key called the embedding key has to be provided to the group members in addition to the original keys, which can lead to a lot of overhead.

The key management approach proposed by Wong Gouda uses key graphs [34]. The method consists of creating a secure group from a basic key management graph mechanism using a star or tree based method. Unfortunately, scalability is not achieved using this mechanism, which is its biggest limitation. A new group keying method proposed by McGrew and Sherman [35] uses one-way functions to compute a tree of keys, called the One-way Function Tree (OFT). In this algorithm, the keys are computed up the tree, from the leaves to the root, enabling this approach to reduce re-keying broadcasts to only about *log(n)* keys. The main limitation of this approach is that it consumes more space.

Wade Trappe et al. proposed a Parametric One Way Function (POWF) [30] based binary tree key management scheme. In this mechanism a session key Ks is attached to the tree above the root node. Each node in the tree is assigned a Key Encrypting KEY (KEK) which serves as an internal key (IK) and each user is assigned to a leaf given the IKs of the nodes from the leaf to the root node. If a balanced tree is complete, i.e., all the leaf nodes have members associated with them, then it is necessary to generate a new layer of nodes when adding new members. When a new user wants to join the group, the keys on the path from this leaf node to the root and also the session key must be changed. These new keys are generated by the GC. If a user departs from the group, then all the keys from leaf node to the root node become invalid. These keys must be updated and distributed using either a bottom up or top down approach. The complexity of the storage is given by $\log_a n+2$ keys for the GC. The amount of storage needed by each individual user is given as $S=a^{L+1}- 1/a-1$ Keys. The computation time is determined largely by the number of multiplications required. The number of multiplications needed to update the KEKs using a bottom up approach is $C_{bu}=a\log_a n - 1$ and using top down approach is $C_{td}= (a-1) (\log_a n(\log_a n+1))/2$. Some of the key management schemes proposed in [36-38] utilize a distributed key management approach which is characterized by having no group controller. The group key can either be generated in a contributory fashion, where all members contribute their own share to the computation of the group key, or be generated by one member.

Volker Roth et al. [39] specified an access control and key management mechanism for mobile agents and proposed a mobile agent structure which supports authentication, security management, and access control for mobile agents. In this mechanism, the data can be moved from one host to another host using mobile agents. A migrating agent can become malicious if its state is corrupted. So this poses a severe problem, hence a server must use some mechanism(s) to protect the mobile agent. The basic mechanism for protecting the content of a mobile agent is encryption together with a digital signature. For this purpose, a tree based agent structure is proposed. It supports hierarchical signing and encryption of agents, key management, agent security policy, and data management. A hierarchical structure is used to manage the decryption key. So the main limitation of their proposal is that a large amount of storage required for storing the secret key.

Yu-Fang Chung et al. [40] presents two novel methods. The first scheme is designed using a top down approach. The second scheme is designed to reduce the size of the public parameters. In the first scheme, the leaf nodes are considered as secret keys and the remaining nodes are considered to be super keys. Super keys are used to derive the secret keys. In this scheme, only the authorized host derives the secret keys that are used to encrypt the confidential files. So the size of the public parameters will grow dramatically with an increasing number of visiting hosts and confidential files. In order to reduce the size of the public parameters, the second scheme is used.

In this scheme, the public parameter is independent of the number of visiting hosts and confidential files. The limitation of this paper is that the system is insecure, because everyone derives the super key and secret key. Additionally, it requires a lot of computation since everyone derives the super key and secret key and thus this is not an optimal method.

Yu Fang Chung et al. [41] described an access control mechanism in a user hierarchy based on using elliptic curve cryptography and a one way hash function. This scheme attempts to derive the secret key of successors efficiently and non-redundantly. The method uses elliptic curve cryptography which has a low computational cost and small key size, thus it provides both efficiency and security. The proposed key management method for controlling dynamic access is a simple and efficient solution for ensuring security in a hierarchical organization. This method allows access by members to data to be classified according to their ranks. Members in a higher-ranked security class can directly access the secret keys of members of lower-ranked classes, but not vice versa. The members can change their secret keys at will, showing that the key generation and public polynomial are flexible. The main drawback is that the computational complexity is greater than their previous method.

Kuo-Hsuan Huang et al. [42] defined an efficient migration strategy for mobile computing in distributed networks. This method secures the accessing relationship between the mobile agent and the host, while minimizing storage space. These properties enable all of the mobile agent to operate efficiently, while providing a secure execution environment for mobile computing. Additionally, the procedures of key generation and operation are very simple; as users with greater accessibility can directly access the decryption key of their subordinate members, but the latter are not allowed to access the decryption key of the former. The drawback is that the user needs to perform additional computations to verify the certificate in this scheme.

Jen-Ho Yang et al. [43] described an ID-based remote mutual authentication with key agreement scheme for mobile devices based on elliptic curve cryptography. This method provides mutual authentication and supports a session key agreement between the user and the server. This scheme is efficient because bilinear-pairing is more expensive than point-multiplication operation when using elliptic curve cryptography. The proposed scheme is constructed based upon an ID-based concept and it utilizes the user's unique identity to compute an authentication key for mutual authentication. Thus, mutual authentication between the user and the server can be accomplished without using public keys. In addition, the users and the server do not need to perform additional computations to verify the other party's certificates providing greater efficiency. However, the system requires more effort to keep the key updated.

In this thesis project a new key management algorithm is proposed and evaluated which focuses on reducing computational complexity, while at the same time increasing security by using a small key.

3.4 Related Work for Intrusion Detection

John F. C. Joseph et al. [53] proposed an autonomous host-based intrusion detection system for detecting malicious sinking behaviour. Their proposed detection system maximizes the detection accuracy by using cross-layer features to define a router's behaviour. For learning and adapting to new attack scenarios and network environments, two machine learning techniques, Support Vector Machines (SVMs) and Fisher Discriminant Analysis (FDA), are used in tandem to combine the higher accuracy of SVM with the faster speed of FDA. Grinblat et al. [54] introduced a time-adaptive support vector machine (TA-SVM), which is a new method for generating adaptive classifiers capable of developing learning concepts, which change as time progresses. The basic idea of a TA-SVM is to use a sequence of classifiers, each one appropriate for a small time window. However, in contrast to other proposals, the learning occurs for all the involved hyperplanes in a global manner. The addition of a new term in the cost function of the set of SVMs (which penalizes the diversity between consecutive classifiers) produces a coupling of the sequence which allows TA-SVM to learn as a single adaptive classifier. The evaluation of their method was carried out using appropriate drifting problems. In particular, they analysed the regularizing effect of modifying the number of classifiers in the sequence or adapting the strength of the coupling.

Khalil El-Khatib [55] proposed a novel hybrid model which efficiently selects the optimal set of features in order to detect IEEE 802.11-specific intrusions. His model for feature selection uses the information gain ratio measure to compute the relevance of each feature and a k-means classifier to select the optimal set of MAC layer features which can improve the accuracy of intrusion detection systems, while reducing the learning time. He studied the impact of the feature set optimization for wireless intrusion detection systems on the performance and learning time of different types of neural network based classifiers. Experimental results with three types of neural network architectures clearly showed that the optimization of a wireless feature set has a significant impact on the efficiency and accuracy of the intrusion detection system. Sooyeon Shin et al. [56] proposed a hierarchical framework which takes into account both intrusion detection and data processing mechanisms, and constructed a set of hierarchical intrusion prevention and detection protocols in their framework. They believe that their hierarchical framework is useful for securing industrial applications and provides two distinct lines of defence. They also believe that their framework makes it easy to apply new detection techniques against future attacks against their intrusion detection protocol.

Xiaowei Yang et al. [57] proposed a Kernel in fuzzy kernel clustering based Fuzzy Support Vector Machine (FSVM) algorithm for classification problems with outliers or noise. The contributions of their research work are as follows. First, a common misunderstanding of Gaussian-function-based kernel fuzzy clustering in the high-dimensional feature space is corrected. Second, an algorithm is proposed for classification problems with outliers or noise. Their experiments were conducted on six benchmarking datasets and four artificial datasets to test the generalization performance of the algorithm. Their results show that their algorithm based on the fuzzy C-Means (FCM) clustering in a high-dimensional feature space that presents the most reasonable membership degrees and is more robust than the other classification algorithms based on FCM clustering. Third, the computational complexity of the KFCM-FSVM algorithm is presented. It should be noted that although this algorithm was tested on binary classification problems, it can also be applied to various classification problems without difficulty. However, they have not yet evaluated large-scale classification problems with outliers or noise.

Kapil Kumar Gupta et al. [58] have addressed the dual problem of accuracy and efficiency for building robust and efficient intrusion detection systems. Their experimental results show that conditional random fields are very effective in improving the attack detection rate and decreasing the false alarm rate (FAR). Having a low FAR is very important for any intrusion detection system. Further, feature selection and implementation of their layered approach significantly reduces the time required to train and test the model. Even though they used a relational data set for their experiments, they showed that the sequence labelling methods, for example conditional random fields can be very effective in detecting attacks and outperforms the other methods known to work well with the relational data. They compared their approach with some well-known methods and found that most of the present methods for intrusion detection fail to reliably detect Remote to Local (R2L) and User to Root (U2R) attacks, while their integrated system can effectively and efficiently detect such attacks, giving an improvement of 34.5 percent for the R2L and 34.8 percent for the U2R attacks.

Serafeim Moustakidis et al. [59] proposed a novel fuzzy decision tree (FDT) (the FDT-support vector machine (SVM) classifier), where node discrimination is implemented via binary SVMs.

The tree structure is determined via a class grouping algorithm, which forms the groups of classes to be separated at each internal node, based on the degree of fuzzy confusion between the classes. In addition, effective feature selection is incorporated within the tree building process, by selecting suitable feature subsets required for the node discriminations individually. FDT-SVM exhibits a number of attractive merits such as enhanced classification accuracy, interpretable hierarchy, and low model complexity.

Carol J. Fung et al. [60] proposed Dirichlet-based trust management to measure the level of trust among IDSs according to their mutual experience. An acquaintance management algorithm is also proposed to allow each IDS to manage its acquaintances according to their trustworthiness. Their approach achieves excellent scalability and is robust against common insider threats, resulting in an effective collaborative intrusion detection network (CIDN). They evaluate their approach based on a simulated CIDN, demonstrating its improved robustness, efficiency, and scalability for collaborative intrusion detection in comparison with other existing models.

S. Umang et al. [61] proposed a novel approach for an enhanced intrusion detection system for malicious nodes, and to protect against attacks on the *ad hoc* on-demand distance vector routing protocol. Their proposed approach employs a method for determining conditions under which a potentially malicious node should be monitored. Apart from identification of malicious nodes, they have been observed that this approach leads to less communication breakage in *ad hoc* routing. Their experimental results demonstrate that the proposed approach can effectively detect malicious nodes.

Shingo Mabu et al. [62] proposed a GNP-based fuzzy class-association-rule mining process with sub-attribute utilization. The proposed classifiers based on the extracted rules can consistently use and combine discrete and continuous attributes in a rule, and efficiently extract many good rules for classification. In anomaly detection, their results show a high detection rate (DR) and reasonable positive false rate (PFR) with no prior machine learning, which is an important advantage of their proposed method.

Jie Liu et al. [63] proposed a framework of combining intrusion detection and continuous authentication in MANETs. In their framework, multimodal biometrics are used for continuous authentication and intrusion detection is modelled as sensors to detect the system's security state. They formulated the whole system as a partially observed Markov decision process considering both system security requirements and resource constraints. They used dynamic programming-based hidden Markov model scheduling algorithms to derive the optimal schemes for both intrusion detection and continuous authentication.

Yuteng Guo et al. [64] proposed a new feature selection method based on Rough Sets and improved genetic algorithms for network intrusion Detection. First, the features are filtered using the Rough Sets theory; then the optimal subset will be found from the remaining feature subset using a genetic algorithm, improved with the population clustering approach, in order to achieve optimized results.

Guanghui Song et al. [65] proposed an intrusion detection method based on a multiple kernel support vector machine. This method can calculates the weights of kernel functions and Lagrange multipliers simultaneously through semi-infinite linear programming, improves detection accuracy by using the radial basis function kernel function with different kernel parameter values, and avoids the need for setting kernel parameters in advance.

Liu Zhiguo et al. [66] proposed a hybrid method combining a rough set and support vector machine together to aid in network intrusion detection. Their detection model includes data reduction by a rough set and network intrusion recognition by support vector machine. A normal support vector machine is compared with the hybrid method of rough set-support vector machine.

Their experimental results show that the detection accuracy of a rough set-support vector machine detection method is greater than that of support vector machine alone.

Lei Li et al. [67] combined a fuzzy support vector machine and a support vector machine based on a binary tree and apply it to intrusion detection. By computer simulations using the KDDCUP99 data set [86], they demonstrated the superiority of their method over a simple multi-class support vector machine. Cao Li-ying et al [68] combined the SVM algorithm and learning vector quantization (LVQ) neural network algorithm and applied it to network intrusion detection. They concluded that: (1) In contrast with back propagation neural networks, the convergence speed of the combined SVM-LVQ model is faster and its detection rate of attacks is significantly higher and error rate is lower. (2) The combined model offers a significantly higher recognition rate than the ordinary method.

Krzysztof Cpałka [69] proposed a new class of neuro-fuzzy systems. He developed a novel reduction algorithm that gradually eliminates inputs, rules, antecedents, and the number of discretization points of integrals in the "center of area" defuzzification method. It then automatically detects and merges similar input and output fuzzy sets. Computer simulations have shown that the accuracy of the system after reduction and merging has not deteriorated, despite the fact that in some cases up to 54% of the parameters and 74% of the inputs were eliminated. The reduction algorithm has been tested using well-known classification benchmarks.

Hu Yan et al. [70] proposed a multi classifier. A Support Vector Machine (SVM) and Artificial Neural Network (ANN) provide a solid base for building this classifier. The top level employs a voting mechanism to identify intrusions, while taking time evolution characteristics into account. In addition, to make the classifier more self-adaptive, an incremental learning module is introduced. The proposed classifier has been successfully applied in oil and gas pipeline intrusion detection systems.

4 CloNe Security Architecture with Key Management

The essential requirement of the security architecture is to translate the security requirements specified by the tenant into concrete resource constraints. Additional (security) requirements may also be provided by the different entities in the architecture. This security goal translation has been integrated in the overall goal translation function described by Bjurling et al. [88] and deployed in the CloNe architecture.

The resource configurations defined at the end of the translation process are deployed by the infrastructure provider, and the translation process is assisted by the different security modules depicted in Figure 6. The various security functions include an access control policy function, an auditing and assurance function, an identity management function, and the central security goal translation function (which forms the backbone of the overall security architecture). The access control policy function is responsible for determining access control policies for each infrastructure service user, and will require a suitable access control policy model to define those policies.

To support the access control policy function, an identity management function has been defined to perform the authentication checks of the varied entities in the infrastructure, and to ensure that only authorized parties are provided access to resources and/or services. The identity management system and central authority can be a single module or they can be two different modules, for this thesis they will be considered to be a single entity. Access control policies can only be successfully deployed in a system when the identities of the participating entities can be ascertained with a high probability. Therefore, a well- defined identity management function is indispensable to a system which wants to implement an access control policy model. Additional desirable features included in this identity management function include a compliance module, a federated identity management module, and an authorization and use profiler management module.

A proposed improvement to the overall CloNe security architecture is to add a backbone key management algorithm (to support the identity management function). Chapter 5 will cover the design and deployment details concerning a key management algorithm. The core algorithms have been customized and deployed in the CloNe infrastructure, and their evaluation and comparisons with other algorithms/mechanisms are described in the respective sections in this and the following chapters.

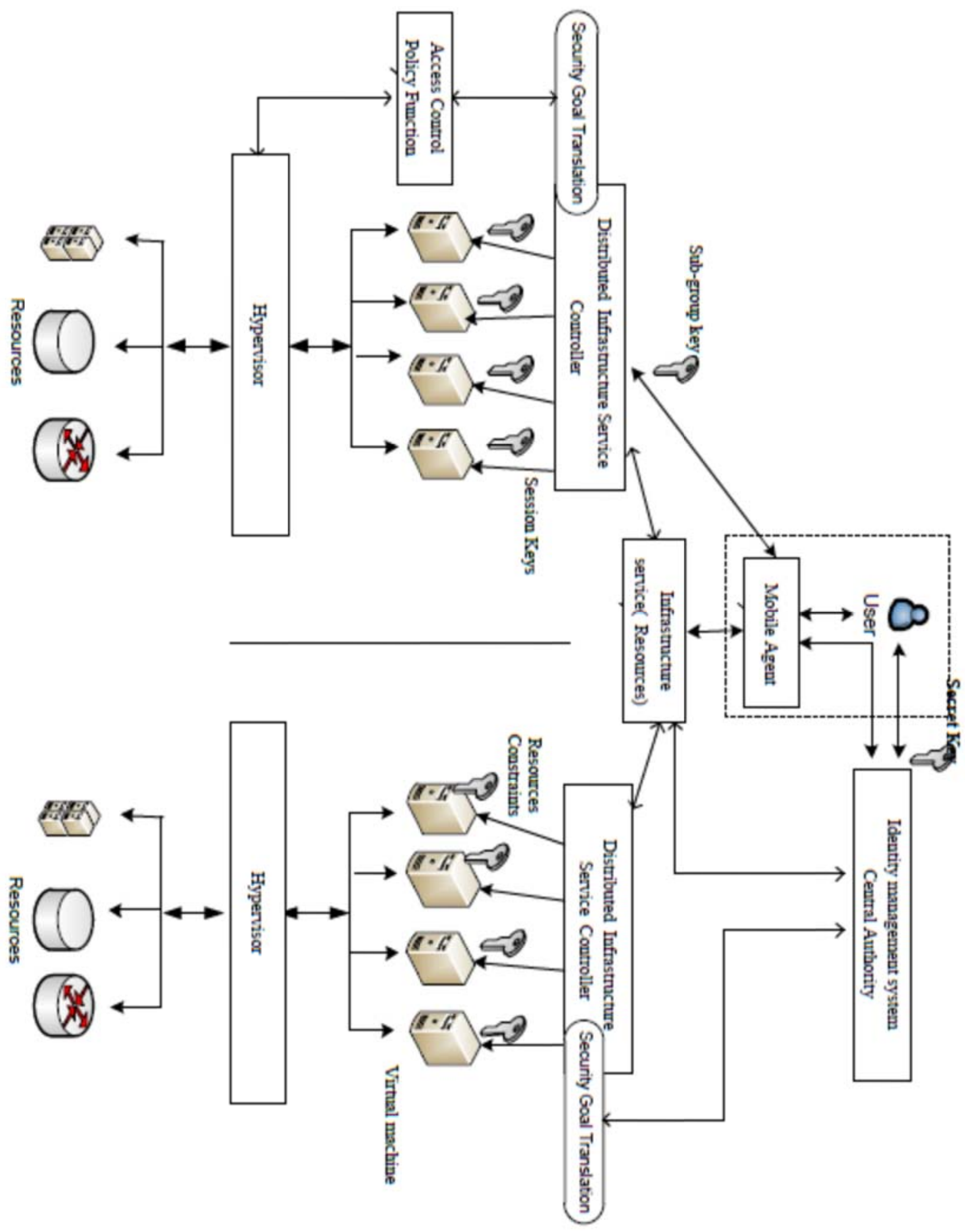

Security Goal Translation
Access Control Policy Function
Distributed Infrastructure Service Controller
Sub-group key
Hypervisor
Resources
Session Keys
Infrastructure service(Resources)
Mobile Agent
User
Secret Key
Identity management system Central Authority
Resources Constraints
Distributed Infrastructure Service Controller
Hypervisor
Resources
Virtual machine
Security Goal Translation

Figure 6: CloNe security architecture with key management mechanism

4.2 Key Management System Architecture

The system architecture of the key management mechanism [92] for CloNe is shown Figure 7 and consists of the following components:

Authorized User	Authorized (Cloud) Users (AU) are the cloud users who are allowed to access the cloud's resources. To access files, search the files, and to decrypt the files a new component called as mobile agent is used in the AU's area.
Mobile Agent	The Mobile Agent (MA) is a program [2] that can roam freely in the Internet environment from local host to other remote hosts in a network. The MA executes tasks assigned by its user. MAs are widely used in distributed systems for distributed computation, data search in remote environments, network management, and work flow systems.
Central Authority	The Central Authority (CA) is part of the identity management function and it controls the access tree structure to maintain the clients as the leaf nodes in the various groups. The CA in its database maintains information about clients, such as the client's private key, userID, and the access level. The sub-group keys maintain the client nodes positions. Each client node stores its private key and public parameters. The storage increases with an increase in the number of nodes.
Servers	Servers are the virtual or physical resources that can be accessed by the mobile agents of the authorized cloud user.

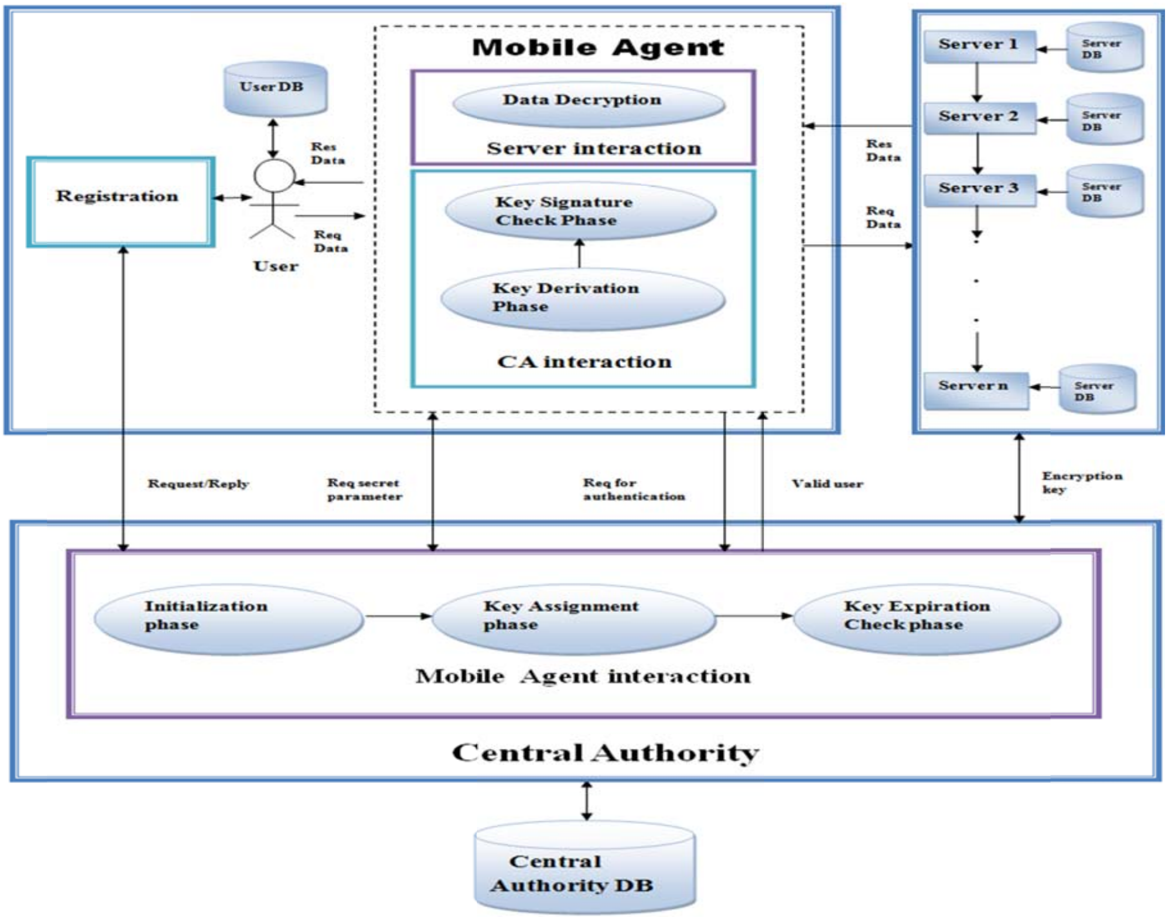


Figure 7: Key Management Architecture [92]

The AU first completes a registration process with the CA and the CA assigns a private key to each user during the registration process. Then the AU sends a file request to access the virtual resources through their MA. The file request contains file name and ID of the user who is sending the request. A MA submits the ID of its user and the file names to be accessed which were obtained by the AU to the CA. Then the CA checks whether the user is an authorized user or not. If the user is an AU then the CA derives the corresponding private key of this user from the user's ID. Using this private key and a base point selected from the elliptic curve, the user's public key is calculated. The CA also checks whether this key has expired or not using the date constraint in the key expiration check phase. If the current date is within the valid period for this key, then the request is sent to the signature creation phase. Now the CA chooses six random numbers to calculate an expiration date for the user's signature using a one way hash chain function during the key assignment phase. Finally, the CA sends the signature to the MA for signature verification and to compute a secret number. The MA computes the secret random number, using this number it also derives its subsequent level keys in the key derivation phase. Finally, the MA computes decryption keys for each file and the file is decrypted using this decryption key.

4.3 Frame work of mobile agent structure

The mobile agent structure consists of two parts: static and mutable [39], [40]. The static part of the agent stores all the static information, i.e., that information that does not change during the agent's lifetime. This information includes security policy, group classification, certificates, and access control keys. To prevent mobile agents from malicious attack by hosts, the server will use a digital signature on the static part to ensure the completeness and authenticity of data. The mutable part stores all the dynamic information that is gathered by a mobile agent. For instance, its status or information collected by the agent can be modified by the visited hosts on the network. Therefore, each visited host must digitally sign the information that has been modified.

Figure 8 shows the mobile agents structure and its components, such as the classes Sctx and Acl. In this structure there are five zipped files: AUC.Zip, AUT.Zip, GCT.Zip, PSG.Zip and CIT.Zip. As indicated in the figure, the files are encrypted by keys Dc1, Dc2, Dc3, Dc4, and Dc5 respectively. On the right side of static part, the nodes S1, S2, S3, and S4 represents different servers with different access control keys. If these servers are authorized to access the specified files, then the decryption keys are copied to the corresponding server folders. S1 is granted access to all the encrypted files, if the keys (Dc1/Dc2/Dc3/Dc4) are copied to the folder of S1. Based upon this figure: S2 is permitted to access the three encrypted files: AUC.Zip, CIT.Zip, and GCT.Zip. Therefore, the keys (Dc1/Dc5/Dc3) are duplicated into the folder of S2. In addition, S3 is only allowed to access the two encrypted files AUC.Zip and AUT.Zip; accordingly, only the keys Dc1 and Dc2 are in the folder of S3. Finally, S4 is permitted to access the first encrypted zipped file, AUC.Zip and thus it will only have a copy of Dc1 to access the designated file.

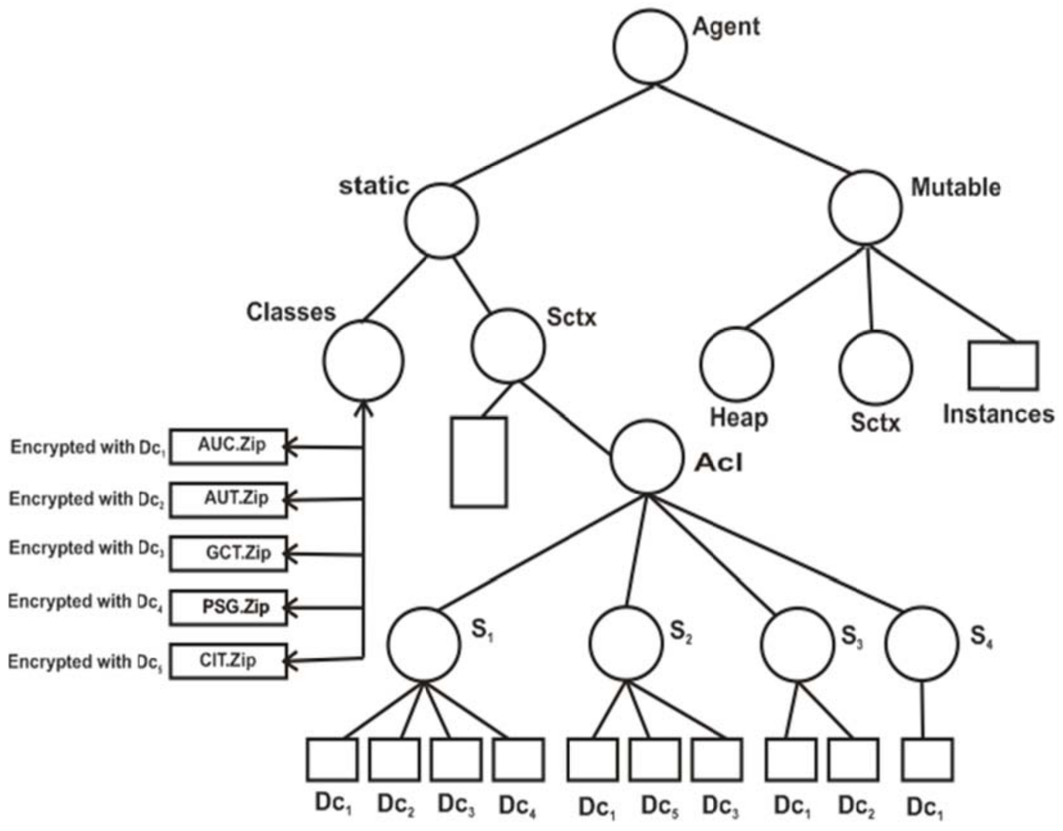


Figure 8: Access control and key management [39]

4.4 Structure of decryption keys for mobile agents

The tree structure shown in Figure 9 is used to explain how to access leaf nodes' confidential files in a hierarchical structure. Consider for example File j is an encrypted confidential file and Ci is the internal node which also represents a user. Here pr_i represents the secret key/private key held by user i. When pr_i has permission to access some encrypted files, it can obtain the confidential files from the corresponding leaf nodes. In this structure taking node C2 as example, C2 holds secret key pr_2 and based on its access rights, it can access File 1, File 2, File 3, File 4, and File 5. However, node C2 is not permitted to access File 6 as the node C2 does not have access rights to access File 6.

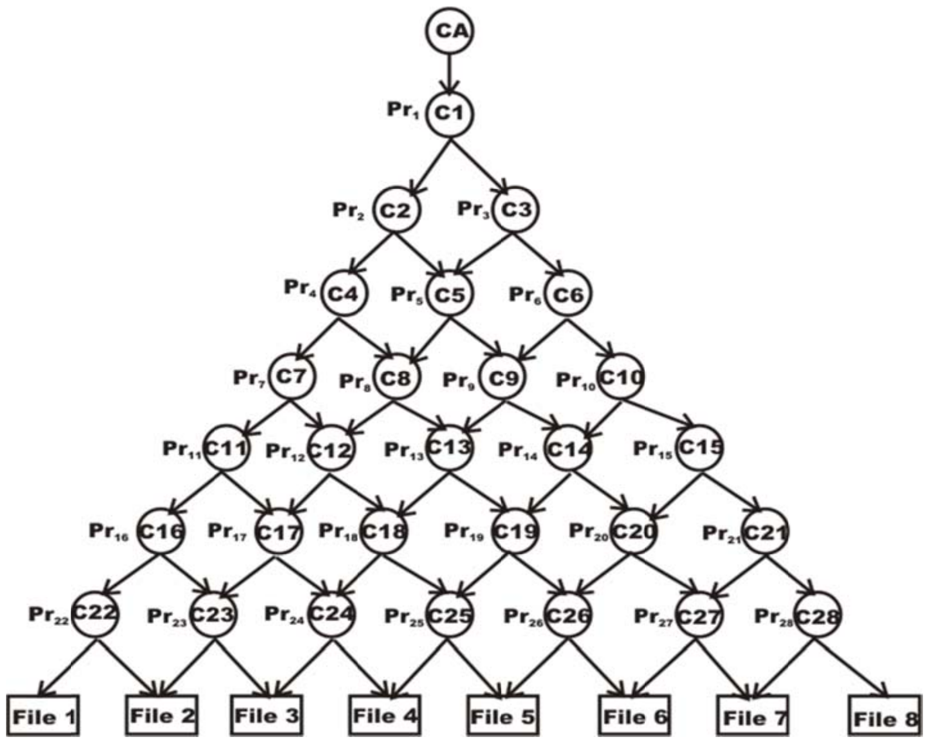


Figure 9: Structure of decryption keys for mobile agent

4.5 Sequence Diagram

Figure 10 shows a sequence diagram. This sequence diagram mainly includes sequence and flow in each process. In the first step the user sends a file request to the mobile agent and the mobile agent will send the ID and file name to the CA. The CA will send the request to the database to retrieve the private key. After retrieving the private key it will invoke the key expiration check phase to check whether the key is valid or not. If it is valid, then it will encrypt the file using the secret key and store it in a database. Then the mobile agent enters into the signature check phase and verifies whether the signature is true or not. If it is true then the mobile agent will request the file from the server and the server will access the required file from the database. After retrieving the file from the database the mobile agent will decrypt the file and display it to the user.

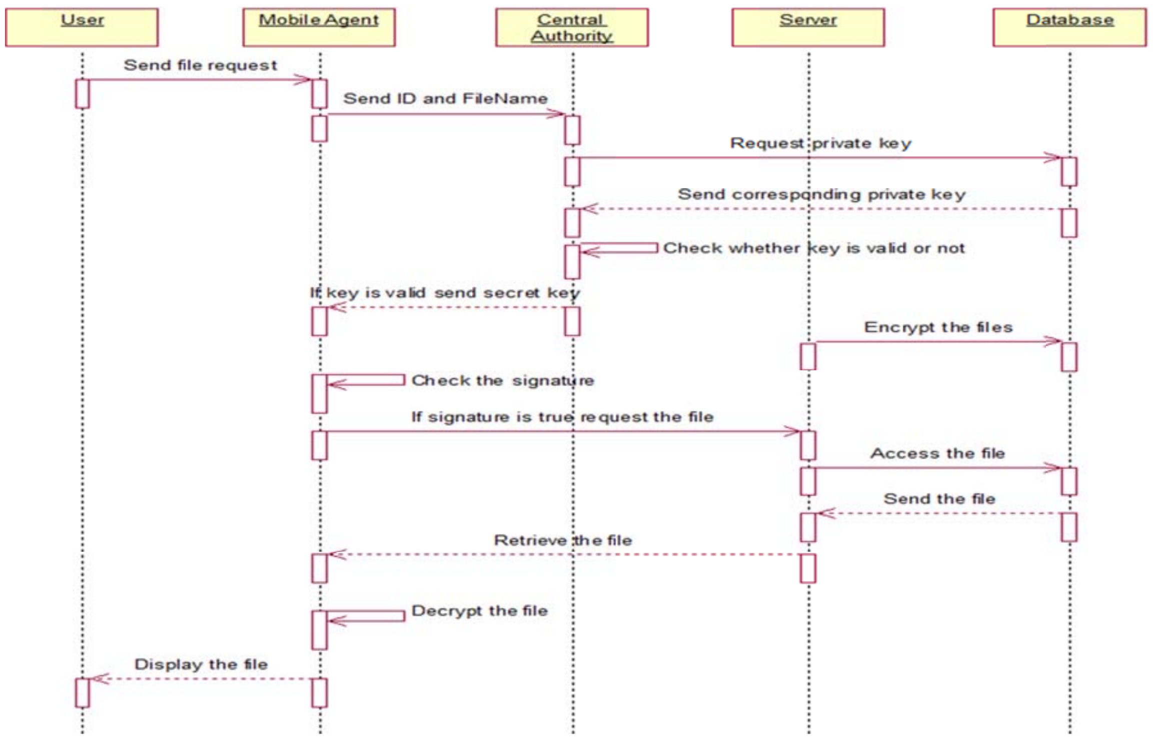


Figure 10: Sequence diagram of key management

5 Implementation of key management module

5.1 Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is an approach to public key cryptography [41], [43] based on the algebraic structure of elliptic curves over finite fields. In the 1980s, Miller and Koblitz introduced elliptic curves into cryptography and Lenstra showed how to use elliptic curves to factor integers. ECC is better than other public key cryptosystems in terms of having lower memory, computational requirements, higher security, and faster implementations [41]. The main advantage of ECC is that a small key of 160 bits is considered as secured as a 1024 bit RSA key. Two families of elliptic curves are prime curves and binary curves. Prime curves are defined over Z_p and binary curves are constructed over $(GF(2^n))$. Prime curves are suitable for implementation in software applications, because the computations do not require extended bit-fiddling operations. Binary curves are more suitable for hardware implementation because the computations require only a few logic gates to build an efficient and fast cryptosystems.

The elliptic curve over Z_p, defined modulo a prime p, is the set of solutions (x, y) to the equation $E_p(a, b)\colon y^2 = x^3 + ax + b (\text{mod } p)$, where $(a, b) \in Z_p$ and $4a^3 + 27b^2 \text{ mod } p \neq 0$.The condition $4a^3 + 27b^2 \text{ mod } p \neq 0$ is necessary to ensure that$y^2 = x^3 + ax + b(\text{mod}p)$has no repeated factors, which means that a finite Abelian group can be defined based on the set $E_p(a, b)$. The definition of an elliptic curve also includes a point at infinity, denoted as O, which is also called the zero point. The point at infinity O is the third point of intersection of any straight line with the curve, so that the points (x, y), $(x, -y)$, and O lie on the straight line. Addition operations are computed over $E_p(a, b)$. The addition rules are:

(1) $+P = P$ and$P + O = P$, where O serves as the additive identity.

(2) $-O = O$.

(3) $P + (-P) = (-P) + P = O$, where $-P$ is the negative of point P.

(4) $(P + Q) + R = P + (Q + R)$.

(5) $P + Q = Q + R$.

For any two points $P = (x_p, y_p)$ and $Q = (x_q, y_q)$over $E_p(a, b)$, the elliptic curve addition operation and which is denoted as$P + Q = R = (x_r, y_r)$ satisfies the following rules:

$$x_r = (\lambda^2 - x_p - x_q) \text{ mod } p, \quad y_r = (\lambda(x_p - x_r) - y_p) \text{ mod } p$$

$$\text{Where } \lambda = \begin{Bmatrix} \left(\frac{y_q - y_p}{x_q - x_p}\right) \text{ mod } p, \text{if } P \neq 0 \\ \left(\frac{3x_p^2 + a}{2y_p}\right) \text{mod } p \text{ , if } p = 0 \end{Bmatrix}$$

5.2 Proposed Key Derivation Protocol

The proposed key derivation protocol (further described in [92]) uses an elliptic curve digital signature algorithm [86], [87] for providing the security in the CA's and each MA's areas. The following steps explain the public key calculation, signature generation, key derivation, key expiration checking, and signature verification phases in detail.

Step 1: CA assigns a private key to each user and defines the elliptic curve $E_p(a,b)$, where $y^2 = x^3 + ax + b \pmod p$ among which a and b must satisfy $4a^3 + 27b^2 \neq 0 \pmod p$ and p is a large prime number.
Step 2: Choose a base point $\mu = (x,y)$ on $E_p(a,b)$, and then user calculates the public key using

$$pu_i = pr_i\, \mu \bmod p$$
$$pu_j = pr_j\, \mu \bmod p$$

Step 3: Next CA randomly chooses six random numbers $\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5$ and γ_6.Then generates $\beta = \beta_1 \parallel \beta_2 \parallel \beta_3 \parallel \beta_4 \parallel \beta_5 \parallel \beta_6$ where $\beta_1 = H^y(\gamma_1)$, $\beta_2 = H^{100-y}(\gamma_2)$, $\beta_3 = H^m(\gamma_3)$, $\beta_4 = H^{12-m}(\gamma_4)$, $\beta_5 = H^d(\gamma_5)$ and $\beta_6 = H^{31-d}(\gamma_6)$. Where y, m and d represents year, month and date respectively, of the date of expiry and " $\parallel$ "is the concatenation operator.
Step 4: Calculate $\alpha = H^{100}(\gamma_1) \parallel H^{100}(\gamma_2) \parallel H^{12}(\gamma_3) \parallel H^{12}(\gamma_4) \parallel H^{31}(\gamma_5) \parallel H^{31}(\gamma_6)$.
Step 5: CA chooses another secret random number λ, and uses the previously calculated parameters α and j to calculate signature date-bound warrant $\alpha \oplus \beta$ and also the following public parameters σ, Z, M and the private key pr_j of userC_j. The calculations are as follows:

$M = \lambda * \mu = (x_1, y_1)$ and
$\sigma = x_1 \bmod p$
$Z = \lambda^{-1} * (H(\alpha \oplus \beta) + pr_i * \sigma) \bmod p$,, $pr_i$is the private key of C_i
$pr_j = H(\lambda, ID_j)$ Where ID_j is the public identifier of C_j.

Step 6: If $C_i \geq C_j$, then C_i can obtain C_j's private key in the following manner. Use public parameter σ, Z, α and j to calculate secret parameter λ , as follows:

$\lambda = Z^{-1} * (H(\alpha \oplus \beta) + pr_i * \sigma) \bmod p$, pr_i is the private key ofC_i.

Step 7: Calculate C_j's private key $pr_j = H(\lambda, ID_j)$.
Step 8: Then calculate $\alpha' = H^{100-y}(\beta_1) \parallel H^y(\beta_2) \parallel H^{12-m}(\beta_3) \parallel H^m(\beta_4) \parallel H^{31-d}(\beta_5) \parallel H^d(\beta_6)$. If α' is not equal to α, then it means the key is already expired.
Step 9: Using the following equation, key signature check is completed by mobile agent.

$k = Z^{-1} \bmod p$
$G_1 = k * H(\alpha \oplus \beta) \bmod p$
$G_2 = (\sigma * k) \bmod p$
$V = (G_1 + G_2 * pr_i)\mu = (x_1, y_1)$

Calculate,

$\omega = x_1 \bmod p$
$M = \lambda * \mu = (x_1, y_1)$ and
$\sigma = x_1 \bmod p$

Step 10: Judge whether ω is equal to σ. If the two values are equal then the signature is true.
Mathematical Proof:

$$\begin{aligned} V &= (G_1 + G_2 * pr_i)\mu = (x_1, y_1) \\ &= \left((k * H(\alpha \oplus \beta)) + (\sigma * k * pr_i)\right)\mu \\ &= (k * (H(\alpha \oplus \beta) + (\sigma * pr_i)))\mu \\ &= \lambda * \mu = M = (x_1, y_1) \end{aligned}$$

5.4 Modified Access Key Hierarchy Based Key Distribution

The access key hierarchy shown in Figure 11 depicts our modified Access Key Hierarchy based key distribution algorithm. The Access Key Hierarchy consists of various levels and in each level various nodes are represented. Each node C_i represents a collection of users who register for that particular node C_i.

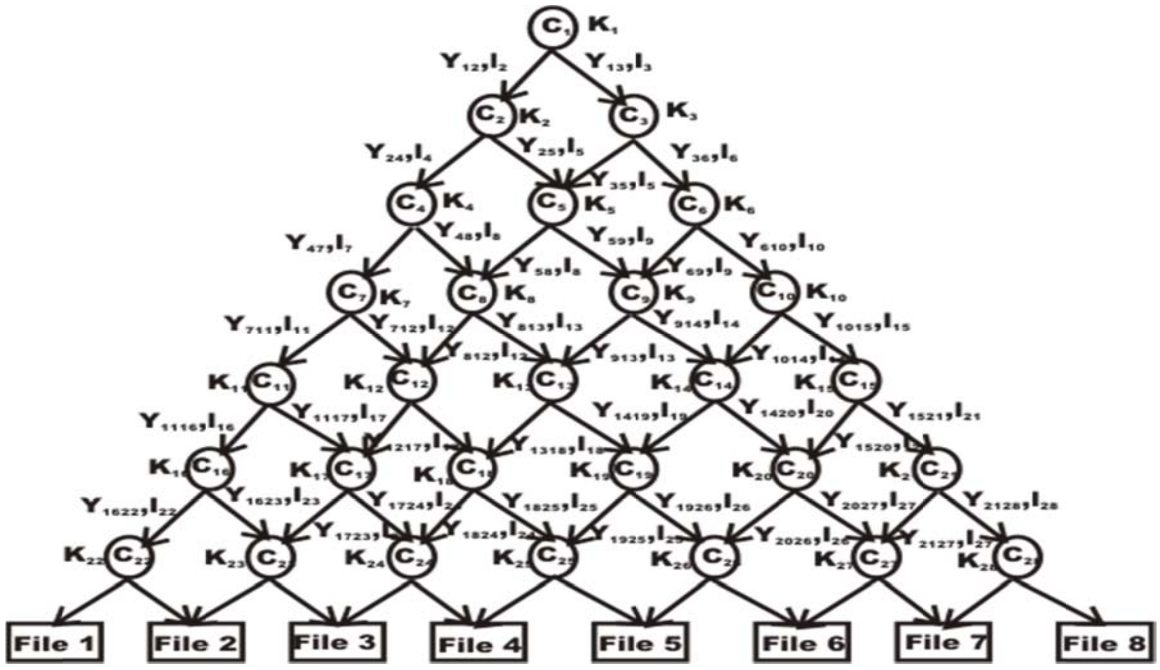


Figure 11: Access key hierarchy

Root node in Access Key Hierarchy is represented as C_i and its subsequent nodes in subsequent levels are represented as C_i+1, $C_i+2,\ldots,C_i+n$ where 'n' is the maximum number of Access levels. The user who registers for the node C_i, he/she will be given full access to access all the files which are attached in the leaf nodes of the Access Key Hierarchy. Similarly, the access level is defined for the Access Key Hierarchy. If the user U_i wants to access the file 1 and he/she has registered for node C_i, he/she has to derive the next level user's secret key K2 using the public parameter Y12 and the public value l2. After that he/she has to derive the secret key K4 using Y24 and l4. Similarly, all the lower level node keys K7, K11, K16 and K22 are derived in order to find the leaf node key value for decrypting the file1.

In general, if the user U_i registers for the node C_i and wants to access all the files in the access hierarchy, he/she has to derive the secret keys of the lower level node keys C_i+1, C_i+2 ,…., C_i+n in order to access all the files File1, File2, …, File n located in the leaf nodes.

The proposed key derivation protocol uses elliptic curve digital signature algorithm [5], [16] for providing the security in CA's and MA's area. The following steps explain the public key calculation, signature generation, key derivation, key expiration check, and signature verification phases in detail.

Step 1: CA assigns a private key to each user and defines the elliptic curve $E_p(a, b)$, where $y^2 = x^3 + ax + b \pmod p$ among which a and b must satisfy $4a^3 + 27b^2 \neq 0 \pmod p$ and p is a large prime number.

Step 2: Choose a base point $\mu = (x, y)$ on $E_p(a, b)$, and then user calculates the public key using

$$pu_i = pr_i\, \mu \bmod p$$
$$pu_j = pr_j\, \mu \bmod p$$

Step 3: Next CA randomly chooses six random numbers $\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5$ and γ_6. Then generates $\beta = \beta_1 \parallel \beta_2 \parallel \beta_3 \parallel \beta_4 \parallel \beta_5 \parallel \beta_6$ where $\beta_1 = H^{y}(\gamma_1)$, $\beta_2 = H^{100-y}(\gamma_2)$, $\beta_3 = H^{m}(\gamma_3)$, $\beta_4 = H^{12-m}(\gamma_4)$, $\beta_5 = H^{d}(\gamma_5)$ and $\beta_6 = H^{31-d}(\gamma_6)$. Where y, m and d represents year, month and date respectively, of the date of expiry and " $\parallel$ "is the concatenation operator.
Step 4: Calculate $\alpha = H^{100}(\gamma_1) \parallel H^{100}(\gamma_2) \parallel H^{12}(\gamma_3) \parallel H^{12}(\gamma_4) \parallel H^{31}(\gamma_5) \parallel H^{31}(\gamma_6)$.
Step 5: CA chooses another secret random number λ, and uses the previously calculated parameters α and j to calculate signature date-bound warrant $\alpha \oplus \beta$ and also the following public parameters σ, Z, M, l, y and the private key pr_j of userC_j. The calculations are as follows:

$$M = \lambda * \mu = (x, y) \text{ and}$$
$$\sigma = (x * H(\alpha \oplus \beta)) \bmod p$$
$$r = x \bmod p \text{ and}$$
$$Z = (\lambda - \sigma * pr_i) \bmod p,$$
$$y = \lambda_j - H(\lambda_i, l_j)$$

, $pr_i$is the private key of C_i .

Step 6: If $C_i \geq C_j$, then C_i can obtain C_j's private key in the following manner:
Use public parameter σ, Z, α and j to calculate secret parameter λ , as follows:
$\lambda = (Z + (\sigma * pr_i)) \bmod p$, pr_i is the private key ofC_i.

Step 7: Calculate C_j's private key pr_j and next level secret keyλ_j. The calculations are as follows:

$$\lambda_j = y + H(\lambda_i, l_j)$$

Step 8: Then calculate $\alpha' = H^{100-y}(\beta_1) \parallel H^{y}(\beta_2) \parallel H^{12-m}(\beta_3) \parallel H^{m}(\beta_4) \parallel H^{31-d}(\beta_5) \parallel H^{d}(\beta_6)$. If α' is not equal to α, then it means the key is already expired.
Step 9: Using the following equation, key signature check is completed by mobile agent.

$$V = (Z + (\sigma * pr_i))\mu = (x, y)$$

Calculate, $\omega = x \bmod p$

$$M = \lambda * \mu = (x, y) \text{ and}$$
$$r = x \bmod p$$

Step 10: Judge whether ω is equal to σ. If the two values are equal then the signature is true.
Mathematical proof:

$$\begin{aligned} V &= (Z + (\sigma * pr_i))\, \mu = (x, y) \\ &= (\lambda - (\sigma * pr_i) + (\sigma * pr_i))\, \mu \\ &= \lambda * \mu \\ &= M \end{aligned}$$

6 Testing of key management module

This chapter describes the experimental test procedure applied to the proposed algorithm. Section 6.1 described the test bed, while section 6.2 describes the CA module, section 6.3 described the client module, and section 6.4 describes the server module.

6.1 Simple experimental cloud test bed

A simple test bed consisting of two computers with similar hardware (an Intel Pentium i7-2720QM (6 MB cache, 2.20 GHz clock) CPU, 8 GB of RAM, an Intel Ultimate-N 6300 (802.11 a/b/g/n) Half Mini Card network interface, and a Seagate 500GB disk drive) were connected with a cross over cable. Each physical computer hosted five virtual machines using Proxmox VE 1.8. Proxmox is an open source virtualization environment. Each VM was running a copy of Microsoft's Windows 7 operating system. We used IPv4 as the communication protocol stack for all VMs and the physical machines. Proxmox VE uses a bridged networking model. These bridges are similar to physical network switches, but implemented in software on the underlying Proxmox VE host. All VMs running on a single underlying host share a single bridge, thus it was as if virtual network cables from each guest were all plugged into a single physical switch. To avoid cross VM communication, VLANs (implementing IEEE 802.1q) are used to separate the networks; thus it is as if each VM were separately connected to the underlying physical system. Each VM can act as a master or a slave depending on the application deployed on this VM. The master node can use resources of one or more slaves at any given time. Each VM has to authenticate every other VM before sharing resources. Even if a VM belongs to the same logical rack, they authenticate each other and their communications are routed through the virtual router whose routing daemon is running on the underlying physical machine.

6.2 Central Authority Module

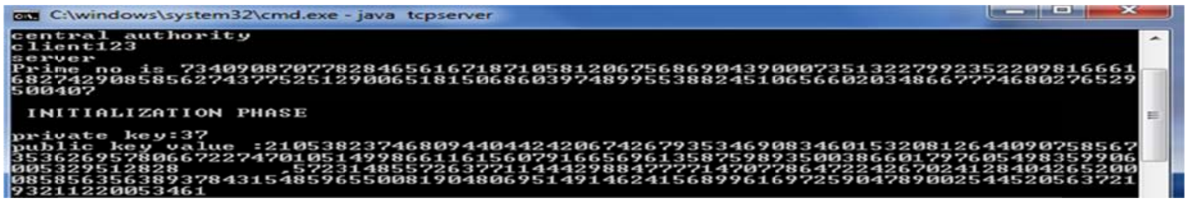
In this section we will describe the initialization, key assignment, and key expiration check phases.

6.2.1 Initialization Phase

The input and output of this phase are:

Input	: The private key of the user and base point.
Output	: The CA calculates the public key value

Figure 12 shows the Initialization phase. In the key derivation protocol, public key calculation is the initialization phase. To calculate the public key, the private key of the user and the elliptic curve base points are used. Initially the private key is converted into a binary string. If the binary string value is 1, doubling and addition operations are performed. If the binary string value is 0, only a doubling operation will be performed. These operations are performed until the end of the binary string. Finally, the public key value is calculated by the CA.



```
central authority
client123
server
Prime no is 7340908707782846561671871058120675686904390007351322799235220981666168274290858562743775251290065181506860397489955388245106566020348667774680276529500407

 INITIALIZATION PHASE

private key:37
public key value :2105382374680944044242067426793534690834601532081264409075856735362695780667227470105149986611615607916656961358759893500386601797605498359906005329512828        ,5723148557263771144429884777714707786472242670241284042652000858563563893784315485965500819048069514914624156899616972590478900254452056372193211220053461
```

Figure 12: Initialization phase

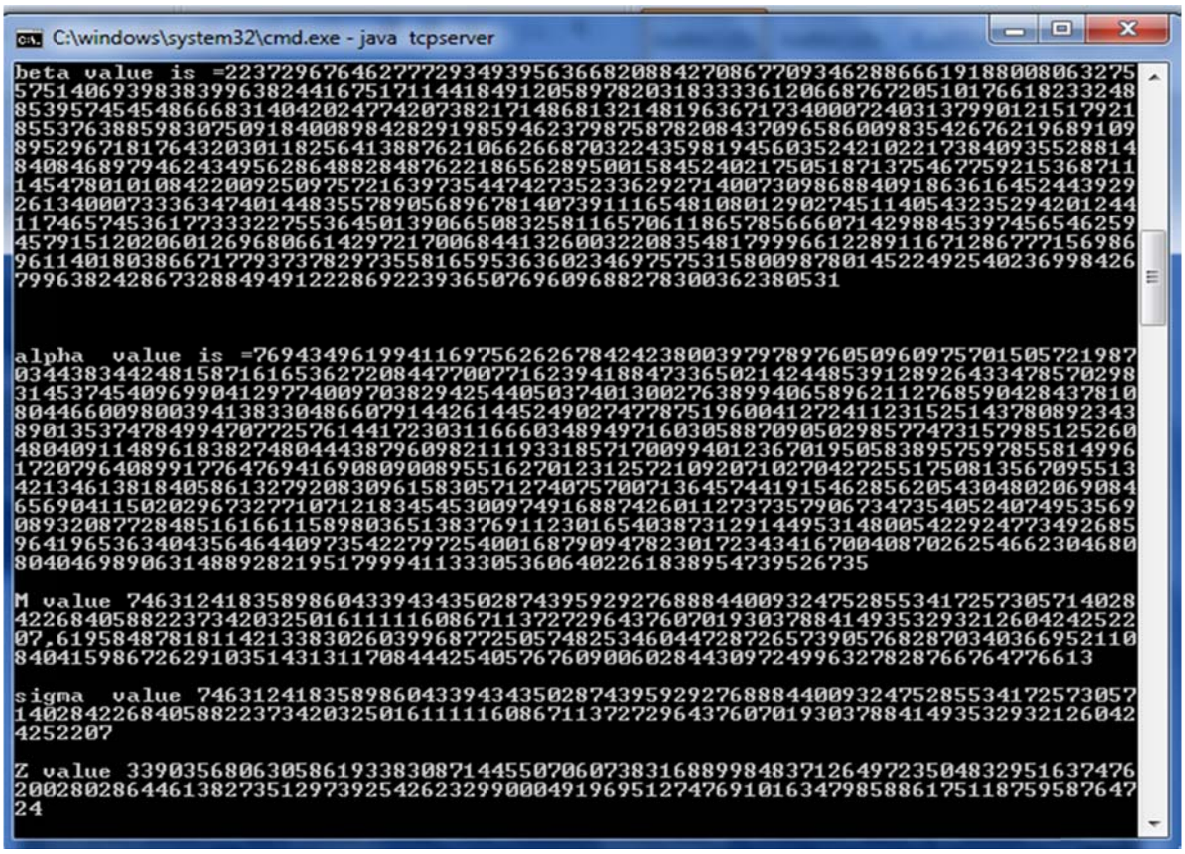
6.2.2 Key Assignment Phase

Figure 13 shows the Key assignment phase. Initially, the CA chooses six random numbers (Gama1, Gama2, … , Gama6) and gets the current year, month, and date. Next the CA performs hash operations using these random numbers and the current date. After that, the results of the hash functions are concatenated and stored as the beta value. In this key assignment phase all the public parameters (sigma, Z, and M) will be calculated.

The input and output of this phase are:

Input : Six random numbers.

Output : CA creates the signature and public parameters.



```
beta value is =22372967646277729349395636682088427086770934628866619188008063275
57514069398383996382441675171144184912058978203183333612066876720510176618233248
85395745454866683140420247742073821714868132148196367173400072403137990121517921
85537638859830750918400898428291985946237987587820843709658600983542676219689109
89529671817643203011825641388762106626687032243598194560352421022173840935528814
84084689794624349562864882848762218656289500158452402175051871375467759215368711
14547801010842200925097572163973544742735233629271400730986884091863616452443929
26134000733363474014483557890568967814073911165481080129027451140543235294201244
11746574536177333227553645013906650832581165706118657856660714298845397456546259
45791512020601269680661429721700684413260032208354817999661228911671286777156986
961140180386671779373782973558165953636023469757531580009878014522492540236998426
79963824286732884949122286922393650769609688278300362380531


alpha  value is =769434961994116975626267842423800397978976050960975701505721987
03443834424815871616536272084477007716239418847336502142448539128926433478570298
31453745409699041297740097038294254405037401300276389940658962112768590428437810
80446600980039413833048660791442614452490274778751960041272411231525143780892343
89013537478499470772576144172303116660348949716030588709050298577473157985125260
48040911489618382748044438796098211193318571700994012367019505838957597855814996
17207964089917764769416908090089551627012312572109207102704272551750813567095513
42134613818405861327920830961583057127407570071364574419154628562054304802069084
65690411502029673277107121834545300974916887426011273735790673473540524074953569
08932087728485161661158980365138376911230165403873129144953148005422924773492685
96419653634043564644097354227972540016879094782301723434167004087026254662304680
8040469890631488928219517999411333053606402261838954739526735

M value 74631241835898604339434350287439592927688844009324752855341725730571402842268405882237342032501611111608671137272964376070193037884149353293212604242522
07,6195848781811421338302603996877250574825346044728726573905768287034036695211084041598672629103514313117084442540576760900602844309724996327828766764776613

sigma  value 7463124183589860433943435028743959292768884400932475285534172573057140284226840588223734203250161111160867113727296437607019303788414935329321260424252207

Z value 3390356806305861933830871445507060738316889984837126497235048329516374762002802864461382735129739254262329900049196951274769101634798588617511875958764724
```

Figure 13: Key assignment phase

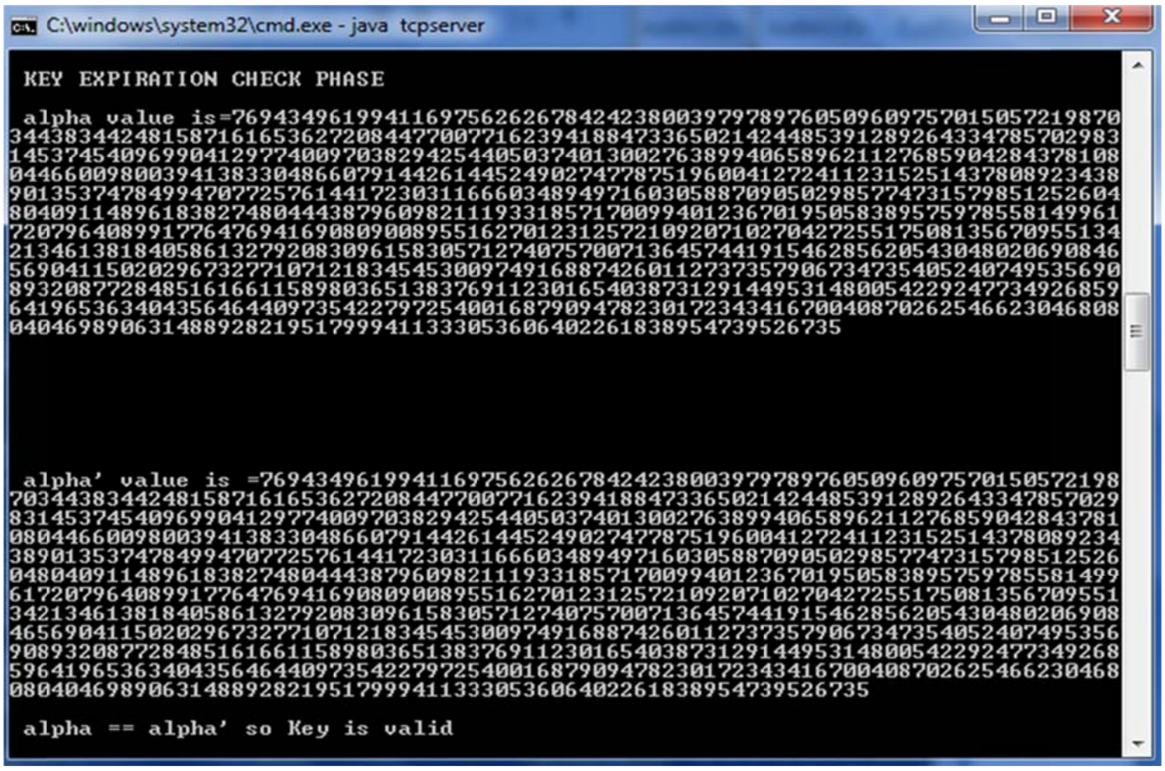
6.2.3 Key Expiration Check Phase

Figure 14 shows the key expiration check phase. In this phase, the CA calculates an alpha value using a hash function. Using the current date, month, year, and beta value, are hashed and then the results of the hashing are concatenated generating the alpha′ value. If the alpha and alpha′

values are equal, then the private key of the user is valid otherwise the key has expired. If the key is valid, then the signature and public parameters are sent to the user. CA sends the encryption key value to server for encrypting the files.

The input and output of this phase are:

Input : Private Key
Output : Check whether the key is valid or not.



```
 KEY EXPIRATION CHECK PHASE

 alpha value is=769434961994116975626267842423800397978976050960975701505721987034438344248158716165362720844770077162394188473365021424485391289264334785702983
14537454096990412977400970382942544050374013002763899406589621127685904284378108
04466009800394138330486607914426144524902747787519600412724112315251437808923438
90135374784994707725761441723031166603489497160305887090502985774731579851252604
80409114896183827480444387960982111933185717009940123670195058389575978558149961
72079640899177647694169080900895516270123125721092071027042725517508135670955134
21346138184058613279208309615830571274075700713645744191546285620543048020690846
56904115020296732771071218345453009749168874260112737357906734735405240749535690
89320877284851616611589803651383769112301654038731291449531480054229247734926859
64196536340435646440973542279725400168790947823017234341670040870262546623046808
0404698906314889282195179994113330536064022618389547395267 35
```

```
 alpha' value is =76943496199411697562626784242380039797897605096097570150572198
70344383442481587161653627208447700771623941884733650214244853912892643347857029
83145374540969904129774009703829425440503740130027638994065896211276859042843781
08044660098003941383304866079144261445249027477875196004127241123152514378089234
38901353747849947077257614417230311666034894971603058870905029857747315798512526
04804091148961838274804443879609821119331857170099401236701950583895759785581499
61720796408991776476941690809008955162701231257210920710270427255175081356709551
34213461381840586132792083096158305712740757007136457441915462856205430480206908
46569041150202967327710712183454530097491688742601127373579067347354052407495356
90893208772848516166115898036513837691123016540387312914495314800542292477349268
59641965363404356464409735422797254001687909478230172343416700408702625466230468
0804046989063148892821951799941133305360640226183895473952673 5
```

```
 alpha == alpha' so Key is valid
```

Figure 14: Key expiration check phase

6.3 Client Module

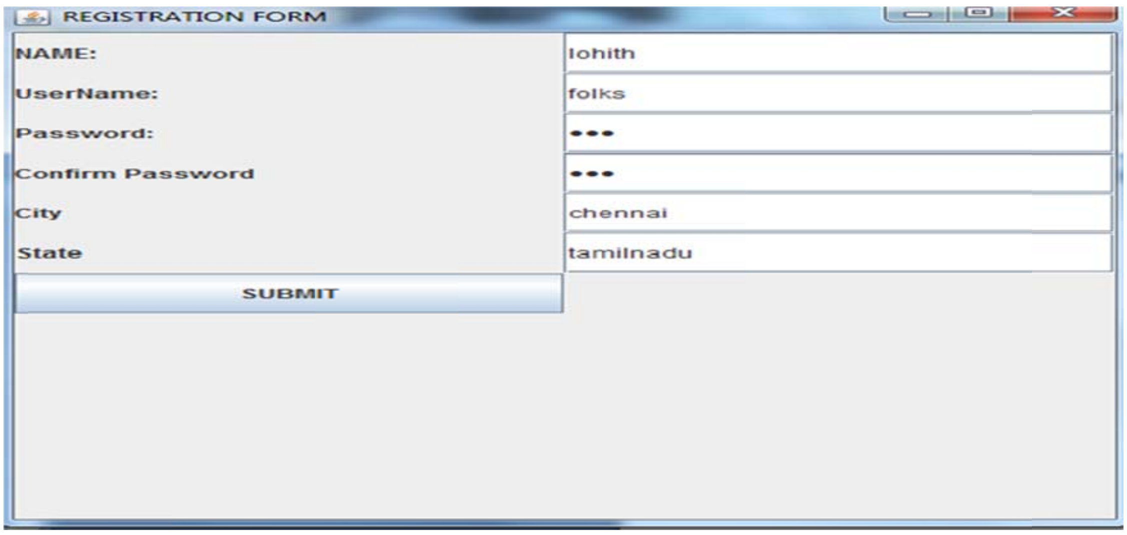
This sections describes three of the client module’s phases: file section, key derivation and signature check, and hierarchical key access phases.

6.3.1 File Selection phase

Whenever the user wants to access any type of file from the server, he/she has to register to access that service. Figure 15 shows the registration form. In this example, in order to register the user enters their user name, password, city, and state. After a successful registration the CA assigns a private key and userID and stores these values in the CA’s database. Figure 16 illustrates the CA login form.

The input and output of this phase are:

Input : Request the file
Output : Chosen file to be accessed by the user



NAME:	lohith
UserName:	folks
Password:	•••
Confirm Password	•••
City	chennai
State	tamilnadu

Figure 15: Registration form

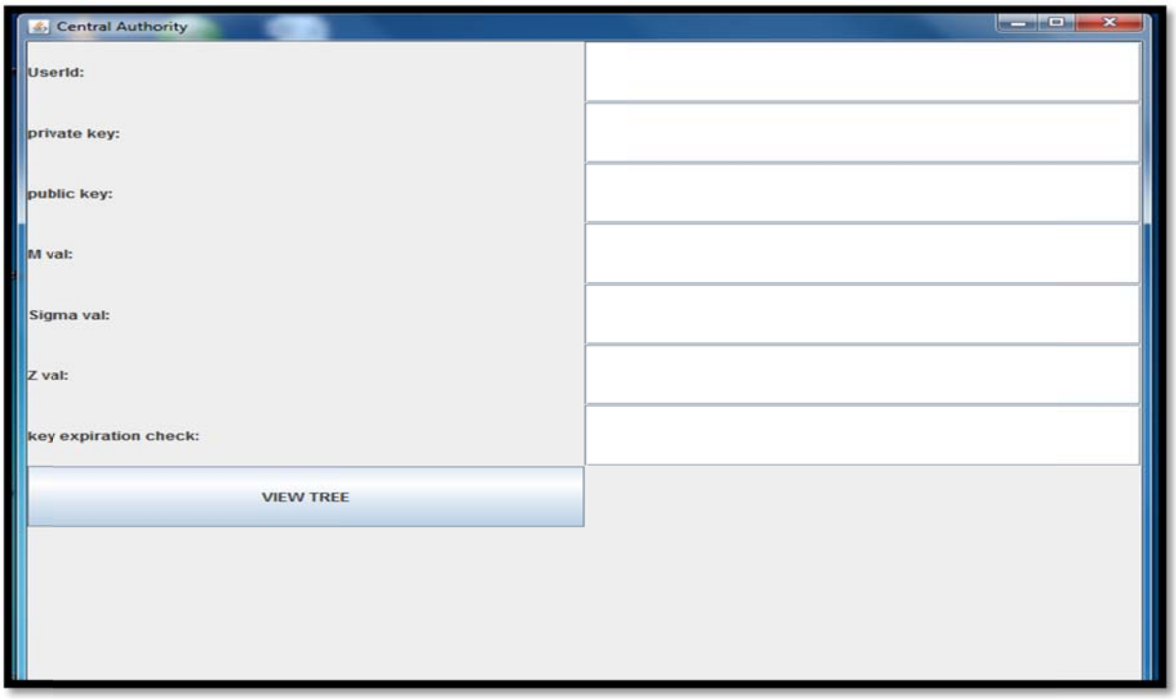
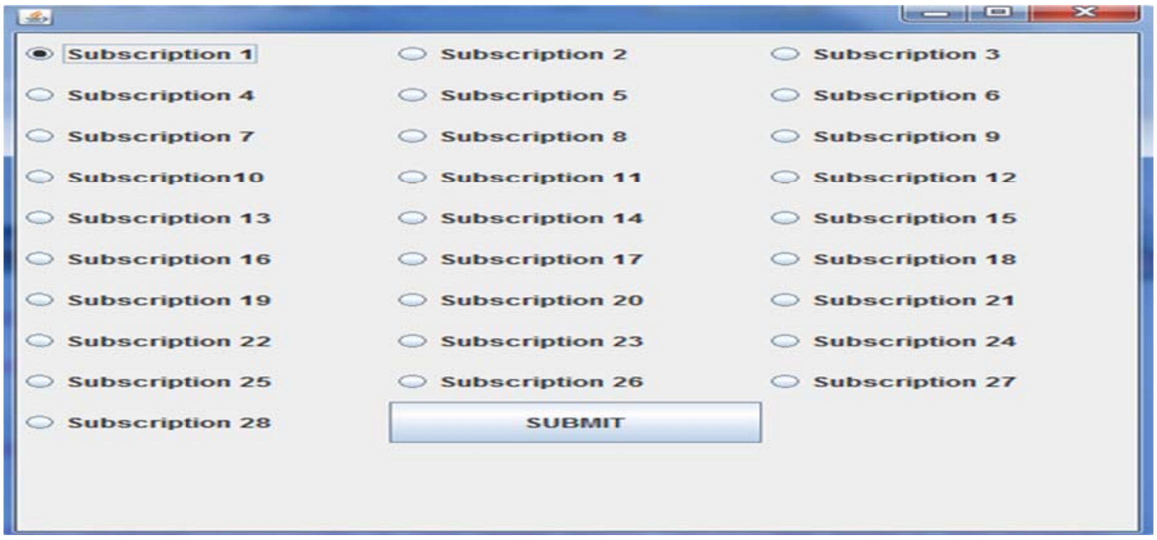


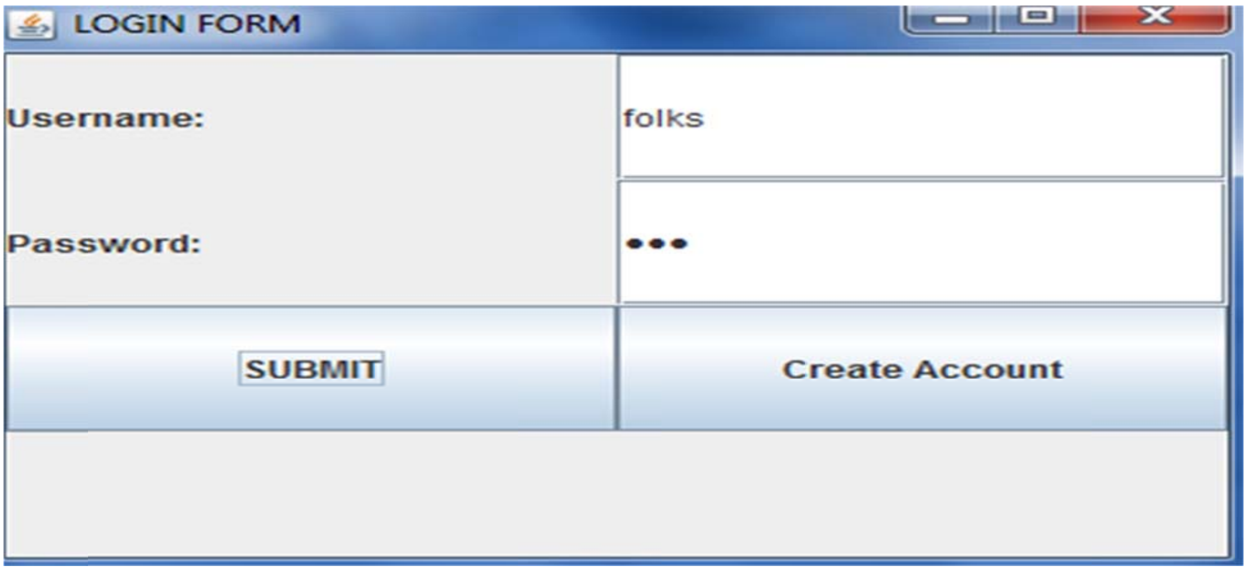
Figure 16: Central authority form

There are various levels of users in the access key hierarchy. As a result, not all users can access all files in the server. Prior to the registration phase, the user had to select a subscription level. Figure 17 shows the User subscription levels and the selected level is used to assign the subsequent access level to the user. Based on this access level he/she can access files from the server. After registration, CA has assigned a private key and userID to each user. Figure 18 shows the Login form. The user enters their username and password. If these matched the values stored by the CA, then the user is logged in to the service at the specified level. After a successful login, the user selects the file to be accessed. Figure 19 shows the file selection form. Using this form the user can select the file they wish to access. The user can view all files which are located in the server, but cannot access all the files. Based on the selected subscription level that the user selected before registering, they can access some of these files. If the user selects a file which they are authorized to access, then he/she can access the file, otherwise the message "you are not authorized to access" will be shown to the user.



● Subscription 1	○ Subscription 2	○ Subscription 3
○ Subscription 4	○ Subscription 5	○ Subscription 6
○ Subscription 7	○ Subscription 8	○ Subscription 9
○ Subscription10	○ Subscription 11	○ Subscription 12
○ Subscription 13	○ Subscription 14	○ Subscription 15
○ Subscription 16	○ Subscription 17	○ Subscription 18
○ Subscription 19	○ Subscription 20	○ Subscription 21
○ Subscription 22	○ Subscription 23	○ Subscription 24
○ Subscription 25	○ Subscription 26	○ Subscription 27
○ Subscription 28	SUBMIT	

Figure 17: User subscription form



LOGIN FORM

Username: folks

Password: •••

Figure 18: User login form

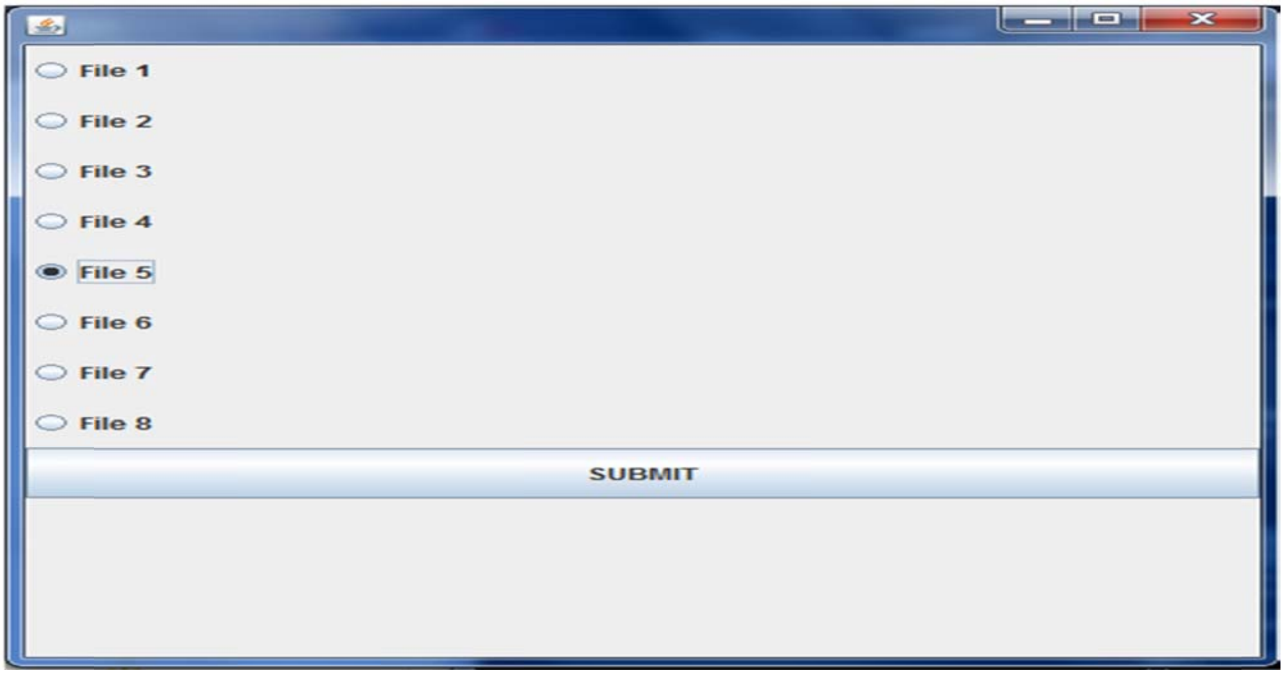


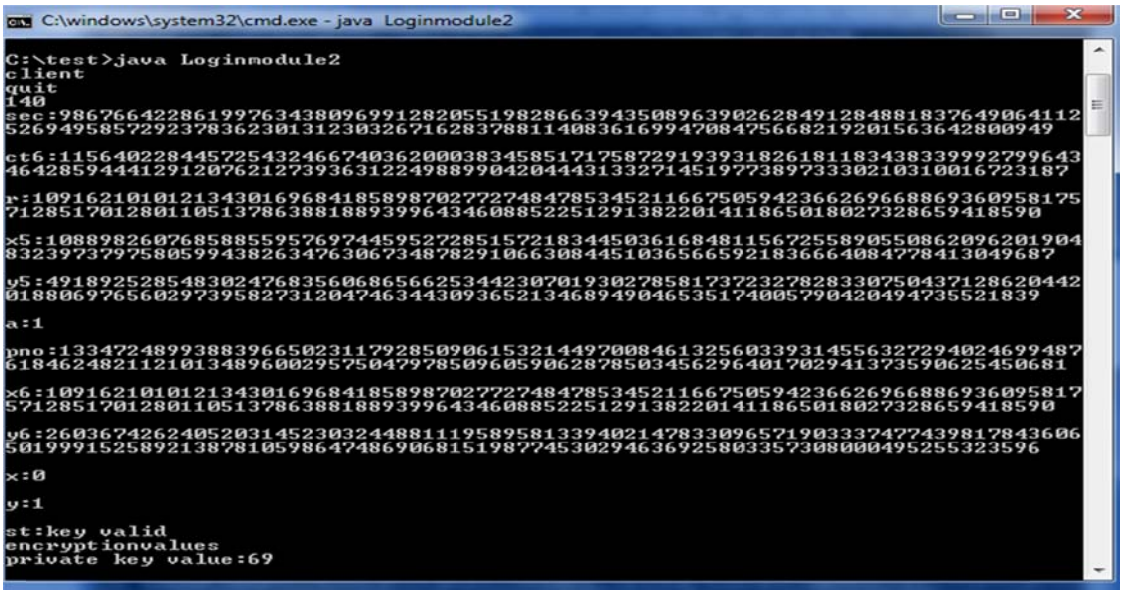
Figure 19: File Selection form

6.3.2 Key Derivation and Signature Check Phase

Figure 20 shows the interaction with the client module. Initially, the user sends a file request to the CA to access a file from the server. The CA verifies whether this user is authorized to access the file from the server or not. If he/she is authorized, then the CA sends the public parameters and the created signature to the user. From these values, the user derives the secret key to decrypt the file. Figure 21 shows the key derivation and signature check phase. In the key derivation sub-phase, the user derives the secret key to access the file using the public parameters and the public value. In the key signature check sub-phase, the signature which was created by the central authority is checked. If the signature is correct, then the user can decrypt the file using the secret key which was derived in the derivation sub-phase.

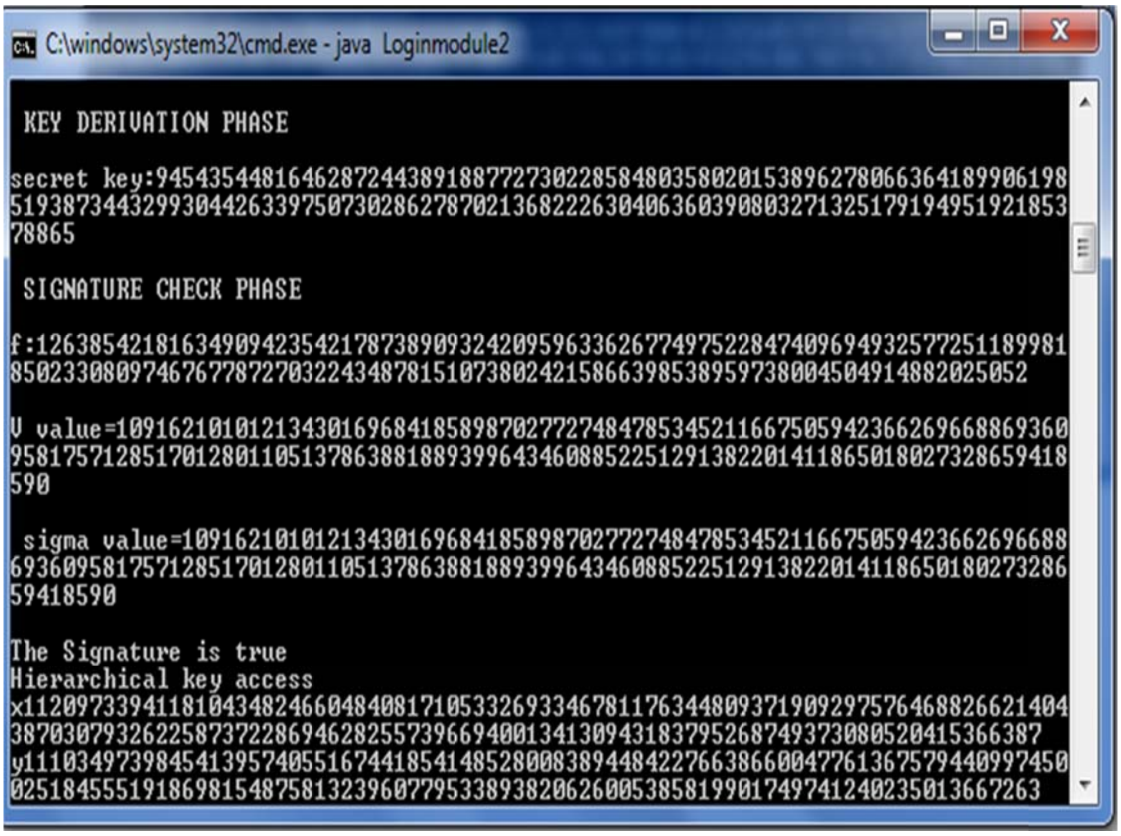
The input and output of this phase are:

Input	: Public parameters and signature.
Output	: Secret key and verify whether the signature is true or not.


C:\windows\system32\cmd.exe - java Loginmodule2

```
C:\test>java Loginmodule2
client
quit
140
sec:9867664228619976343809699128205519828663943508963902628491284881837649064112
526949585729237836230131230326716283788114083616994708475668219201563642800949

ct6:1156402284457254324667403620003834585171758729193931826181183438339992799643
46428594441291207621273936312249889904204443133271451977389733302103100167231 87

r:1091621010121343016968418589870277274847853452116675059423662696688693609581 75
7128517012801105137863881889399643460885225129138220141118650180273286594185 90

x5:108898260768588559576974459527285157218344503616848115672558905508620962019 04
83239737975805994382634763067348782910663084451036566592183666408477841304968 7

y5:491892528548302476835606865662534423070193027858173723278283307504371286204 42
01880697656029739582731204746344309365213468949046535174005790420494735521839

a:1

pno:133472489938839665023117928509061532144970084613256033931455632729402469948 7
61846248211210134896002957504797850960590628785034562964017029413735906254506 81

x6:109162101012134301696841858987027727484785345211667505942366269668869360958 17
57128517012801105137863881889399643460885225129138220141118650180273286594185 90

y6:260367426240520314523032448811195895813394021478330965719033374774398178436 06
50199915258921387810598647486906815198774530294636925803357308000495255323596

x:0

y:1

st:key valid
encryptionvalues
private key value:69
```

Figure 20: Client module


C:\windows\system32\cmd.exe - java Loginmodule2

```
 KEY DERIVATION PHASE

secret key:9454354481646287244389188772730228584803580201538962780663641899061 98
51938734432993044263397507302862787021368222630406360390803271325179194951921853
78865

 SIGNATURE CHECK PHASE

f:12638542181634909423542178738909324209596336267749752284740969493257725118998 1
850233080974676778727032243487815107380242158663985389597380045049148820250 52

V value=10916210101213430169684185898702772748478534521166750594236626966886936 0
95817571285170128011051378638818893996434608852251291382201411186501802732865941 8
590

 sigma value=109162101012134301696841858987027727484785345211667505942366269668 8
69360958175712851701280110513786388188939964346088522512913822014111865018027328 6
59418590

The Signature is true
Hierarchical key access
x11209733941181043482466048408171053326933467811763448093719092975764688266214 04
38703079326225873722869462825573966940013413094318379526874937308052041536638 7
y11103497398454139574055167441854148528008389448422766386600477613675794409974 50
02518455519186981548758132396077953389382062600538581990174974124023501366726 3
```

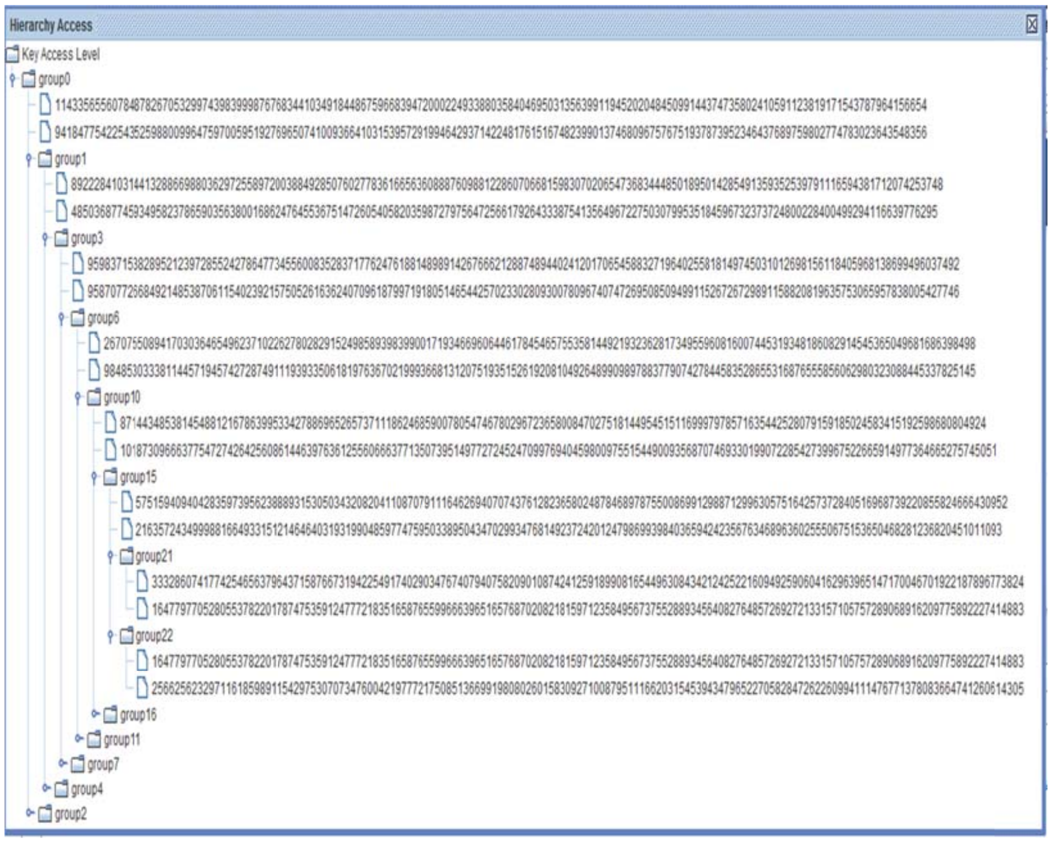
Figure 21: Key derivation and key signature check phase

6.3.3 Hierarchical Key Access Phase

Figure 22 depicts hierarchical key access tree. Initially the user derives the secret key of their own level using the secret key parameter, public parameters, and public key. From this secret key, the user derives all the lower level secret keys and finally finds out the decryption key of the selected file.

The input and output of this phase are:

Input : Secret key parameter for particular level
Output : All leaf level nodes and decryption keys



```
Hierarchy Access
Key Access Level
 group0
   1143356556078487826705329974398399987676834410349184486759668394720002249338803584046950313563991194520204845099144374735802410591123819171543787964156654
   9418477542254352598800996475970059519276965074100936641031539572919946429371422481761516748239901374680967576751937873952346437689759802774783023643548356
   group1
     8922284103144132886698803629725589720038849285076027783616656360888760988122860706681598307020654736834448501895014285491359352539791116594381712074253748
     485036877459349582378659035638001686247645536751472605405820359872797564725661792643338754135649672275030799535184596732373724800228400499294116639776295
     group3
       95983715382895212397285524278647734556008352837177624761881489891426766621288748944024120170654588327196402558181497450310126981561118405968138699496037492
       9587077266849214853870611540239215750526163624070961879971918051465442570233028093007809674074726950850949911526726729891158820819635753065957838005427746
       group6
         2670755089417030364654962371022627802829152498589398399001719346696064461784546575535814492193236281734955960816007445319348186082914545365049681686398498
         9848530333811445719457427287491119393350618197636702199936681312075193515261920810492648990989788377907427844583528655316876555856062980323088445337825145
         group10
           871443485381454881216786399533427886965265737111862468590078054746780296723658008470275181449545151169997978571635442528079159185024583415192598680804924
           10187309666377547274264256086144639763612556066637713507395149772724524709976940459800975515449009356870746933019907228542739967522665914977364665275745051
           group15
             5751594094042835973956238889315305034320820411087079111646269407074376128236580248784689787550086991298871299630575164257372840516968739220855824666430952
             216357243499988166493315121464640319319904859774759503389504347029934768149237242012479869939840365942423567634689636025550675153650468281236820451011093
             group21
               3332860741774254656379643715876673194225491740290347674079407582090108742412591899081654496308434212425221609492590604162963965147170046701922187896773824
               1647797705280553782201787475359124777218351658765599666396516576870208218159712358495673755288934564082764857269272133157105757289068916209775892227414883
             group22
               1647797705280553782201787475359124777218351658765599666396516576870208218159712358495673755288934564082764857269272133157105757289068916209775892227414883
               2566256232971161859891154297530707347600421977721750851366991980802601583092710087951116620315453943479652270582847262260994111476771378083664741260614305
           group16
         group11
       group7
     group4
   group2
```

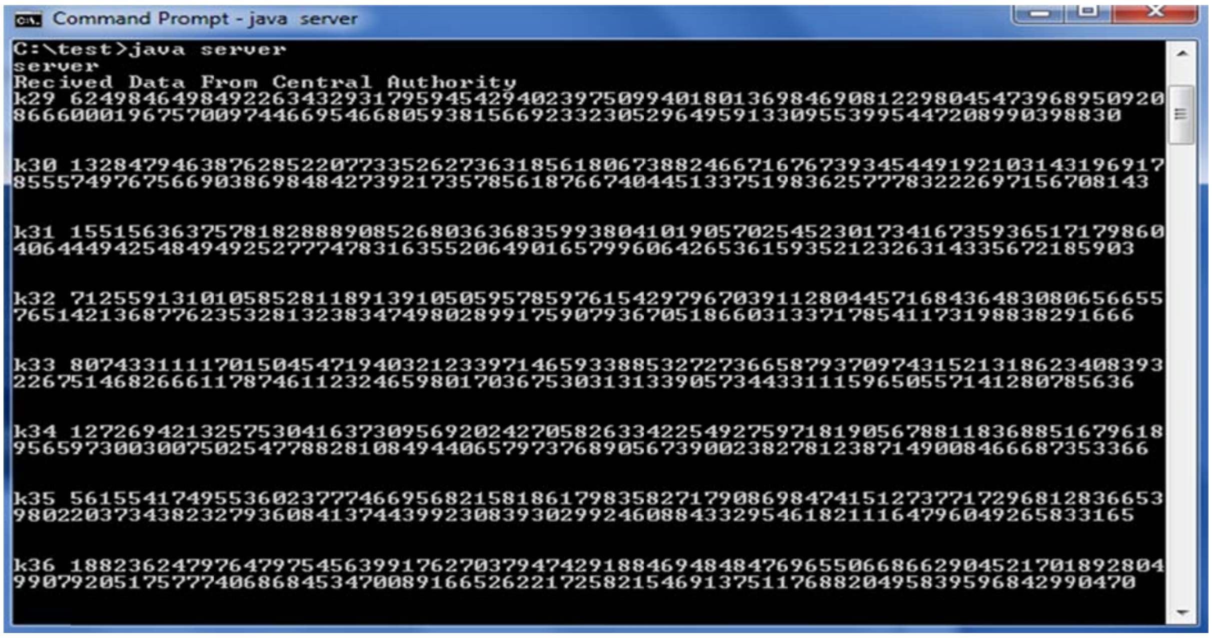
Figure 22: Hierarchical key access tree

6.4 Server Module

Figure 23 shows interaction with the server module. The server receives the encryption key which is sent by the CA. Using this encryption key the server encrypts all the files. When the server receives the requested file from the CA, the server sends the encrypted file to the user.

The input and output of this phase are:

Input : Encryption key derived from central authority.
Output : Encrypt the file using encryption key.



```
C:\test>java server
server
Recived Data From Central Authority
k29 6249846498492263432931795945429402397509940180136984690812298045473968950920
86660001967570097446695466805938156692332305296495913309553995447208990398830


k30 1328479463876285220773352627363185618067388246671676739345449192103143196917
8555749767566903869848427392173578561876674044513375198362577783222697156708143


k31 1551563637578182888908526803636835993804101905702545230173416735936517179860
406444942548494925277747831635520649016579960642653615935212326314335672185903


k32 7125591310105852811891391050595785976154297967039112804457168436483080656655
765142136877623532813238347498028991759079367051866031337178541173198838291666


k33 8074331111701504547194032123397146593388532727366587937097431521318623408393
226751468266611787461123246598017036753031313390573443311159650557141280785636


k34 1272694213257530416373095692024270582633422549275971819056788118368851679618
95659730030075025477882810849440657973768905673900238278123871490084666687353366


k35 5615541749553602377746695682158186179835827179086984741512737717296812836653
980220373438232793608413744399230839302992460884332954618211164796049265833165


k36 1882362479764797545639917627037947429188469484847696550668662904521701892804
990792051757774068684534700891665262217258215469137511768820495839596842990470
```

Figure 23: Server module

7 Results and Discussions

This chapter describes the performance of the proposed key derivation protocol and the proposed hierarchical access key performance.

7.1 Proposed key derivation protocol performance

Table 2 summarizes the overall computational complexity of the proposed key derivation protocol and compare it with two other existing key derivation protocols DCHK [89] and EDHK [90]. The notations used for comparisons are defined as: *Exp* defines the number of exponential operations, *Inv* is the number of inverse operations to be performed in various algorithm by using extended Euclidean algorithms [91], *Mod* is the modular division operations used in various algorithms, *H*, *M*, *PM,* and *PA* denotes Hashing, Multiplications, Point Multiplications, and Point Additions (respectively). *PM* alone requires 'n' point doubling operation if λ is equal to an n-bit binary number and it would take (a-1) point additions if the value λ consists of 'a' number of 1's. Hence, point multiplication takes more computation time than the normal traditional multiplication. However, *PM* takes less computation time than exponential operations. Therefore, the proposed approach takes less computation time than DCHK since it takes only 2*PM*, whereas DCHK takes 4 *Exp*. Similarly the proposed approach takes less computation time than EDHK since it takes 2*PM* whereas EDHK takes 5*PM* in the CA. Moreover, the proposed approach takes less computation time on the user's side compared with EDHK since it does require one less *PA* and one less *PM* on the user's side (although it requires 3 more *Mod* and 3 more *M*).

Table 2: Computational complexity of existing key derivation algorithms and the proposed algorithm

	DCHK	EDHK	Proposed approach
Central Authority (CA)	$(4Exp + 2Inv + 8Mod + 20H + 5M)$	$(5PM + 1Inv + 2Mod + 20H + 2M)$	$(2PM + 1Inv + 3Mod + 20H + 2M)$
User	$(2H + 2M + 3Mod + 3Exp)$	$(2H + 2M + 1Mod + 1Inv + 1PA + 2PM)$	$(1H + 5M + 4Mod + 1Inv + 1PM)$

Table 3 shows the computation times measured in milliseconds of the various functions. When compared with all the other functions, the *Mod* operation takes less computation time and *PM* takes the most computation time. Table 4 shows the measured computation time in nanoseconds for comparison. It is evident from the values shown in this table that the computation time for the proposed algorithm is better both for the GC and MA than the other two algorithms.

Table 3: Computation time complexities of various functions

	16 bit (ms)	**32 bit (ms)**	**64 bit (ms)**
Mod	*2.8*	*3.1*	*3.2*
Hash	*2.9*	*3.9*	*4.6*
Point Addition	*4.2*	*4.4*	*4.7*
Multiplication	*4.8*	*5.0*	*5.4*
Inverse	*5.3*	*5.4*	*6.0*
Point Multiplication	*14.2*	*29.0*	*36.4*

Table 4: Computation time complexities of various key phases with proposed algorithm with all times in nanoseconds (ns)

		Key Initialization	Key Assignment	Key Derivation	Key Expiration	Key Signature
DCHK(ns)	**16 bit**	1600124417	13277579684	3400416	27563131	28080974
	32 bit	1291163383	16310765178	4918147	28520380	50840465
	64 bit	1777496945	18439429736	6754401	30561050	62443371
EDHK(ns)	**16 bit**	1256696766	10892414530	3295533	19374416	22122763
	32 bit	1252036943	14927868579	4689462	19499124	33617340
	64 bit	1546020918	17259713769	6313585	19654424	43734521
Proposed Method(ns)	**16 bit**	122509119	10892414429	3131462	19295886	11202652
	32 bit	1252036785	12674881216	4343105	19498114	25135795
	64 bit	1539056537	17074694181	5913585	19653439	36659748

The graphical results shown in Figure 24 and Figure 25 are used to compare the key derivation and key signature check phase computation time of the proposed method with the existing methods. It compares the results obtained from the proposed approach and an RSA and an ECC based approach, specifically: DCHK and EDHK. From Figure 24 it is observed that when the key is 64 bits, the key derivation time is 5.9136ms with this proposed approach for updating all the keys from the root node to the leaf node, which is better than the other two existing schemes. Moreover, if the number of files to be accessed by a mobile agent increases, then the computation time gradually increases. However, it is still less than the time required with the other two existing approaches. The result shown in Figure 25 compares the key signature check phase computation time of the proposed method with the other two existing methods. When the key size is 64 bits, the computation time of the key check signature is found to be 36.6577ms in the proposed approach, which is better than with the other two existing schemes. As noted earlier, Table 4 summarizes the overall computation complexity of our AKH based Key Derivation protocol and two other well-known key derivation protocols: DCHK [89] and EDHK [90].

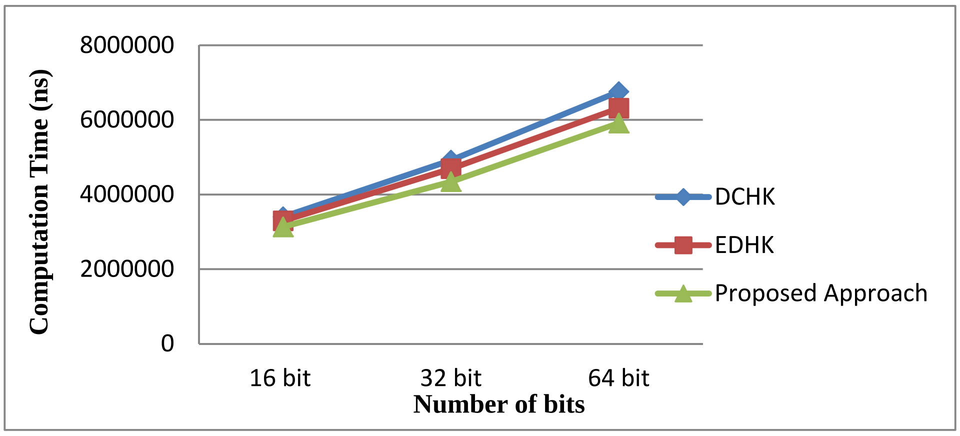


Figure 24: Computation time of existing key derivation algorithms with AKH

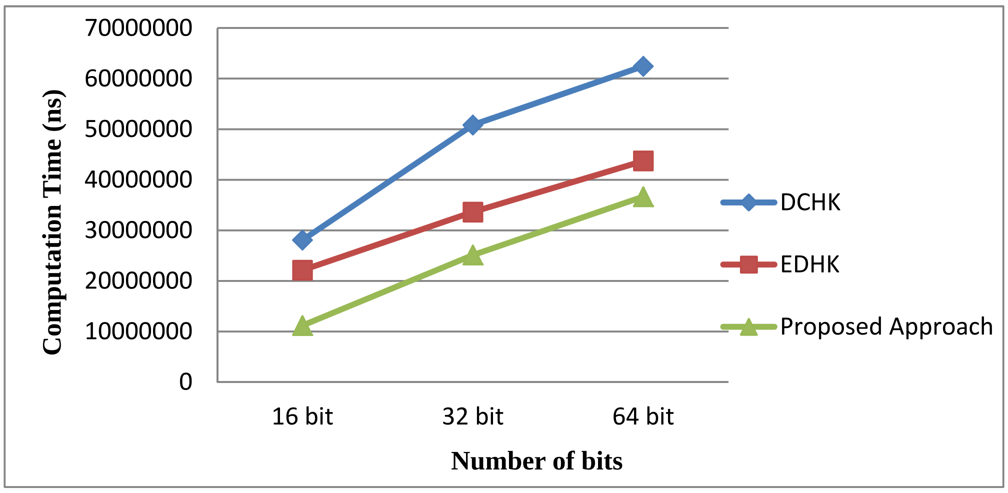


Figure 25: Computation time for key signature check phase with AKH

7.2 Proposed hierarchical key access performance

Table 5 and Table 6 compare the proposed key management algorithm with two other key management protocols. It is evident that the computation time for the proposed algorithm is less both for the GC and each of the MAs than the other two algorithms.

Table 5: Computation complexities of existing key derivation algorithms compared with the proposed algorithm

	DCHK	EDHK	Proposed approach
Central Authority (CA)	$(4Exp + 2Inv + 8Mod + 20H + 5M)$	$(5PM + 4Mod + 20H + 2M)$	$(2PM + 1Inv + 3\,Mod + 20H + 2M)$
User	$(2H + 2M + 3Mod + 3Exp)$	$(2M + 1Mod + 1PM)$	$(1H + 5M + 4Mod + 1Inv + 1PM)$

Table 6: Computation time complexities of various key phases with AKH

	DCHK(ns)			**EDHK(ns)**			**Proposed Method(ns)**		
	16 bit	**32 bit**	**64 bit**	**16 bit**	**32 bit**	**64 bit**	**16 bit**	**32 bit**	**64 bit**
Key Initialization	1600124417	1291163383	1777496945	1256696766	1252036943	1546020918	122509119	1252036785	1539056537
Key Assignment	13277579684	16310765178	18439429736	10892414530	14927868579	17259713769	9892414429	10674881216	14074694181
Key Derivation	3400416	4918147	6754401	3295533	4689462	6313585	2685404	3916589	5411205
Key Expiration	27563131	28520380	30561050	19374416	19499124	19654424	19295886	19498114	19653439
Key Signature	28080974	50840465	62443371	22122763	33617340	43734521	7780589	16089842	25897368

The graphical results shown in Figure 26 and Figure 27 compare the key derivation and key signature check phase computation time of the proposed method with two existing methods. It compares the results obtained from our proposed key derivation protocol with the existing RSA based and ECC based approaches, specifically DCHK and EDHK. From Figure 26 it is observed that when the key is 64 bits, the key derivation time is 5.4112ms in the proposed approach for updating all the keys from the root node to the leaf node, which is better than either of the two other existing schemes. Moreover, if the number of files to be accessed by a mobile agent increases, then the computation time gradually increases. However, it remains less than with the other two existing approaches. The results shown in Figure 27 compare the key signature check phase computation time of the proposed method with the two existing methods. Our proposed key check signature phase when the key size is 64 bits takes 25.8973ms, which is faster than the other two existing schemes.

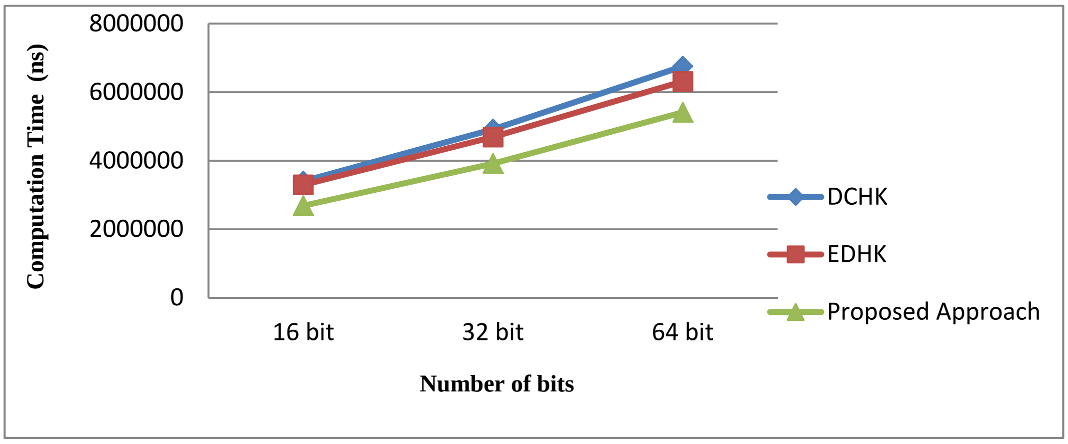


Figure 26: Computation complexities of existing key derivation algorithms with AKH

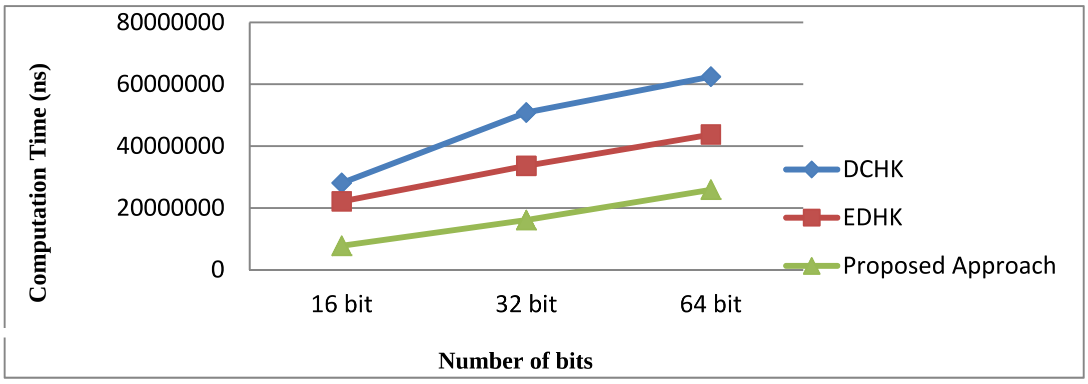


Figure 27: Computation complexities of various key phases with AKH

Table 6 summarizes the overall computational complexity of the proposed hierarchical access key as a function of the level in the hierarchy. The right column in this table shows the measured computation time in nanoseconds to compute an access key. From this table, we observe that when the user registers at level 1 it will take 599.5ms to derive the secret key for decrypting a file. The level 1 computation time is much greater than for all the other levels, because all the groups come under this level. If the user registers at level 2, then the number of files that can be accessed by the user is decreased when compared with level 1 and computation time at the level 2 to compute the access key is 433.95ms. Similarly at level 3 it takes 427.79ms to find the decryption key. Likewise, level 4 takes 417.55ms, level 5 takes 398.33ms, level 6 takes 385.18ms, and level 7 takes 375.87ms to compute the decryption key. Clearly it takes more time to compute this key as we go up in the hierarchy toward the root.

Table 7: Computation time for various levels (in nanoseconds)

	Computation Time (ns)
Level 1	599505728
Level 2	433955921
Level 3	427798821
Level 4	417550403
Level 5	398336207
Level 6	385184262
Level 7	375873060

8 Intrusion Detection

An intrusion detection system with a “Genetic Algorithm based Feature Selection Algorithm for Effective Intrusion Detection in Cloud Networks” has been developed as part of this thesis. In this intrusion detection model a newly proposed genetic based feature selection algorithm and an existing Fuzzy SVM for effective classification are combined. The main advantage of the proposed genetic algorithm based feature selection algorithm is that it improves the detection accuracy of the fuzzy SVM classifier. The genetic algorithm filters most of the less relevant features and thereby provides minimum and required number of features to fuzzy SVM, which helps to reduce the classification time and increases the classification accuracy. The proposed IDS aims to detect the plausible attacks on the CloNe architecture with an acceptable success rate. This research is described in a research paper submitted to KDCloud 2012 “The 2nd International Workshop on Knowledge Discovery Using Cloud and Distributed Computing Platforms”. A copy of this paper is included in appendix B.

9 Conclusions and future work

This chapter concludes the thesis offering a conclusion and suggesting some future research. The chapter concludes with some required reflections on social, economic, and ethical considerations.

9.1 Conclusion

In this thesis a set of new techniques for securing different resources present in a cloud network environment which are not addressed in already existing CloNe security architecture have been proposed. For this purpose a new CloNe security architecture and a security system architecture that explains the newly proposed key management technique using mobile agents for providing effective security through an access control mechanism has been designed and implemented. This proposed key management scheme uses a hierarchical key management approach so that it is possible to provide fast and secure key update. Moreover, this hierarchical approach allows us to perform key updates effectively when a member joins or a member leaves. The main advantages of this proposed scheme are an increase in performance through a reduction in key computation and updating time and an increase in security based on the techniques used for key computations. From the experiments carried out in this project, it has been observed that the proposed key management scheme enhances the security and increases the availability and integrity of the system.

Further, to strengthen the CloNe security architecture, a newly proposed genetic algorithm based feature selection technique has been employed for effective feature selection. In addition, an existing classification algorithm called Fuzzy SVM has been used on the data set for effective classification. The main advantage of the proposed genetic based feature selection algorithm is that it reduces the number of features and hence decreases the classification time. Moreover, it improves the detection accuracy of the fuzzy SVM classifier by minimizing the conflicting rules that may confuse the classifier. The main advantages provided by this intrusion detection system are a reduction in false positives and increase in security.

9.2 Future work

In the future the users of the CloNe architecture must be able to use services such as OpenID for authorisation. This would enable customers to migrate from one cloud service provider to another cloud service provider more easily. This can be complimented with a new key management proposal to reduce the communication complexity in terms of the number of keys to be sent from GC to group members. Further extensions to this work are to devise reliable batch rekeying techniques for reducing the rekeying cost for batch join and batch leave. Reliability should be introduced in multicast key management scheme in order to compute the new group key from the previous group key values received from the GC when the new group key is lost. Another possible future task is to develop a new key signature algorithm to provide secure mobile communication in a distributed environment, since the schemes proposed and implemented in this thesis focused on centralized key management schemes.

9.3 Required reflections

Research which proposes new solutions in the areas of security and privacy potentially has many social and ethical implications. However, during the course of this project we have not encountered any major issues that have new social, ethical, or environmental implications. The research implications are essentially the same as for all the existing work in each of the areas.

For the economics of the project, I would like to express my gratitude to the Fraunhofer-AISEC for its funding of my research through the European commission "Scalable and Adaptable Internet Solutions", SAIL Project (FP7-ICCT-2009-5-257448).

References

1. SAIL project website. http://www.sail-project.eu/, 2010.
2. T. Leva, J. Goncalves, R.J. Ferreira, and J. Myrberger, "D-2.1 (D-A.1) Description of project wide scenarios and use cases". *Technical report, FP7-ICT-2009-5 257448-SAIL/D.A.1 Deliverable*, 2011.
3. WARD Consortium, "Virtualisation approach", Concept. Technical report, ICT-4WARD project, Deliverable D3.1.1, 2009.
4. J. Smith and R. Nair, "The architecture of virtual machines," *Computer*, vol. 38, no. 5, pp. 32–38, 2005.
5. P. Barham, B. Dragovic, K. Fraser, S. Hand, A. Ho, R. Neugebauer, I. Pratt, and A. Wareld, "XEN and the art of virtualization," *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pp. 164–177, 2003.
6. R.P. Goldberg, "Survey of Virtual Machine Research," *IEEE Computer*, pp.34–45, 1974.
7. J. S. Turner, "A proposed architecture for the GENI backbone platform in *Proceedings of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems*, New York, USA, pp. 1-10, 2006.
8. A. Edwards, A. Fischer, A. Lain, "A new approach to networking within virtualized infrastructures". Diverter, *Tech. Rep. HPL-2009-231*, HP Laboratories, 2009.
9. M. Agrawal, S. R. Bailey, A. Greenberg, J. Pastor, P. Sebos, S. Seshan, K. V. D. Merwe, and J. Yates, "RouterFarm: Towards a Dynamic, Manageable Network Edge," in *In Proc. ACM SIGCOMM Workshop on Internet Network Management INM*, 2006.
10. A.Edwards, A. Fischer, A. Lain, "A new approach to networking within Virtualized infrastructures". Diverter, Tech. Rep. HPL-2009-231, HP Laboratories, 2009.
11. Amazon elastic block store. *http://aws.amazon.com/ebs/*, 2010.
12. Intel virtualization. http://www.intel.com/technology/ virtualization/, 2010.
13. KDD CUP Dataset, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999.
14. How server and network virtualization make data centre more dynamic. *http://www.cisco.com/web/CN/products/products_netsol/ucs/pdf/featured_content_dynamic_02.pdf*, 2009.
15. D. Fraser, "The Canadian response to the USA Patriot Act. Security Privacy", *IEEE 5(5)*, 66-68, 2007.
16. Directive 2002/58/EC of the European Parliament and of the Council of 12 july 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (directive on privacy and electronic communications). In: Ocial Journal of the European Union, L201, pp. 0037-0047, 2002.
17. Amazon virtual private cloud. *http://aws.amazon.com/vpc/*, 2010.
18. G. Pallis and A. Vakali, "Insight and perspectives for content delivery networks," *Communications of the ACM*, vol. 49, no. 1, pp. 101–106, 2006.
19. N.M. Mosharaf, Kabir Chowdhury, and Raouf Boutaba, 'A survey of network virtualization', Computer Netwworks, vol. 54, no. 5, pp. 862–876, April 2010, DOI:10.1016/j.comnet.2009.10.017.
20. A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford, "In VINI veritas: realistic and controlled network experimentation," *SIGCOMM Computer Commununications Review*, vol. 36, no. 4, pp. 3–14, 2006.
21. N. Feamster, L. Gao, and J. Rexford, "How to lease the internet in your spare time". *SIGCOMM Computer Commununications. Review, vol.* 37 no. 1, pp. 61-64, 2007.

22. G. Schaffrath, C. Werle, P. Papadimitriou, A. Feldmann, R. Bless, A. Greenhalgh, A. Wundsam, M. Kind, O. Maennel, and L. Mathy, “Network virtualization architecture: proposal and initial prototype,” in *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, New York, NY, USA, pp. 63–72, 2009.
23. FEDERICA: Federated E-infrastructure Dedicated to European Researchers Innovating in Computing network Architectures. http://www.fp7-federica.eu/, 2010.
24. Y. Wang, E. Keller, B. Biskeborn, J. van der Merwe, and J. Rexford, “Virtual routers on the move: live router migration as a network-management primitive,” *SIGCOMM Computer Commununications Review*, vol. 38, no. 4, pp. 231–242, 2008.
25. G. Brunette, R. Mogul. Security guidance for critical areas of focus in cloud computing v3.0. Cloud Security Alliance, *https://cloudsecurityalliance.org/wp-content/themes/csa/guidance-download-box.php,* 2011.
26. W. Streitberger, A. Ruppel. "Cloud computing security - protection goals, taxonomy, market review". *Tech. rep.*,Institute for Secure Information Technology SIT, 2010.
27. G. Koslovski, P.V. Primet, and A.S Charao, VXDL: Virtual Resources and Interconnection Networks Description Language. In *GridNets 2008*, 2008.
28. M. Li, R. Poovendran, and C. Berenstein, “Design of secure multicast key management schemes with communication budget constraint,” *Communications Letters, IEEE*, vol. 6, no. 3, pp. 108 –110, 2002.
29. R. Poovendran and J.S. Baras, "An information-theoretic approach for design and analysis of rooted-tree-based multicast key management schemes", *IEEE Transactions on Information Theory. vol. 47*, pp. 2824–2834, 2001.
30. W. Trappe, J. Song, R. Poovendran, and K. J. R. Liu, “Key management and distribution for secure multimedia multicast,” *IEEE Transactions on Multimedia*, vol. 5, no. 4, pp. 544 – 557, 2003.
31. M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols", IEEE Transactions on Information Theory. vol. 42, no. 2, pp. 529 –542, pp. 1996
32. Lihao Xu, Cheng Huang, "Computation-efficient multicast key distribution", *IEEE Transactions on Parallel and Distributed Systems. vol. 19*, pp .1-10, 2008.
33. W.Trappe, J. Song, R. Poovendran, K.J.R. Liu, "Key distribution for secure multimedia multicasts via data embedding", *IEEE International Conference on Acoustics, Speech, and Signal Processing, Maryland University, College Park, MD* ,2001.
34. C.Wong, M.Gouda, and S.Lam, “Secure group communications using key graphs”, *IEEE/ACM Transactions on Networking. vol. 8*, pp.16-30, 2008.
35. D.A. McGrew and A.T. Sherman,”Key establishment in large dynamic groups using one-way function trees”, *Cryptographic Technologies Group, TIS Labs at Network Associates*, pp.6-12, 1998.
36. Hongsong Shi, Mingxing He, “A communication-efficient key agreement Protocol in Ad hoc Networks”, *IEEE International Conference on Wireless Networks, Communications and Mobile Computing*, China, 2005.
37. H.Seba, F.Tigrine and H.Kheddouci, “A tree-based group key agreement scheme for secure multicast increasing efficiency of rekeying in leave operation”, *IEEE Symposium on Computers and Communications*, France, 2009.
38. Mahalingam Ramkumar, "The subset keys and identity tickets (SKIT) key distribution scheme". *IEEE Transactions on Information Forensics And Security. vol. 5,* pp.39-51.
39. V. Roth and M. Jalali, Concepts and architecture of a security-centric mobile agent server.*"In Proc. Fifth International Symposium on Autonomous Decentralized Systems (ISADS 2001), pages 435-442*", Texas, U.S.A., 2001.
40. Y. Chung, T. Chen, C. Liu, T. Wang, "Efficient Hierarchical Key Management Sc*heme for Access C*ontrol in the Mobile Agent". *AINA Workshops*, 650-655, 2008.

41. Y. Chung, H. Lee, F. Lai, T. Chen: "Access control in user hierarchy based on elliptic curve cryptosystem". *Inf. Sci. vol. 178(1):* 230-243, 2008.
42. K. Huang, Y. Chung, C. Liu, F. Lai, T. Chen, "Efficient migration for mobile computing in distributed networks". *Computer Standards & Interfaces Vol. 31*(1): 40-47, 2009.
43. J. Yang, C. Chang: "An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem". *Computers & Security vol. 28(3-4)*, 138-143, 2009.
44. P. Schoo, V. Fusenig, V. Souza, M. Melo, P. Murray, H. Debar, H. Medhioub, and D. Zeghlache, "Challenges for cloud networking security," *in Mobile Networks and Management, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Springer Berlin Heidelberg, 2010.
45. Paul Murray et al. D-5.2 (D-D.1) Cloud Network Architecture Description. Report, 2011.
46. Volker Fusenig and Ayush Sharma. Security architecture for cloud networking. To appear in Proceedings of the 2012 International Conference on Networking and Computing, ICNC 2012, IEEE Computer Society, 2012.
47. Open Security Architecture cloud computing pattern, February 2011. [Online]. Available: http://www.opensecurityarchitecture.org/cms/library/patternlandscape/251-pattern-cloudcomputing.
48. "Amazon Virtual Private Cloud". http://aws.amazon.com/ec2/, 2011
49. CloudSecurity Alliance GRC Stack, https://cloudsecurityalliance.org/research/grc-stack/, 2011.
50. M.A. Harrison, W.L. Ruzzo, and J.D. Ullman, "Protection in Operating Systems". *Communication, ACM vol. 19 no. 8,* pp. 461-471, 1976.
51. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, C. E. Youman. "Role-Based Access Control Models". *IEEE Computer vol. 29 no. 2*, pp. 38-47, 1996.
52. A.A. Kalam, R.E. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miège, C. Saurel, and G. Trouessin, "Organization Based Access Control". *IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, Lake Come, Italy, 2003.
53. J. F. C. Joseph, B.S. Lee, A. Das, and B.C. Seet, "Cross-Layer Detection of Sinking Behavior in Wireless Ad Hoc Networks Using SVM and FDA", *IEEE Transactions on Dependable Sec. Comput. vol. 8, no. 2,* pp. 233-245, 2011.
54. L. Zhiguo; K. Jincui; L. Yuan, "A hybrid method of rough set and support vector machine in network intrusion detection," *Signal Processing Systems (ICSPS), 2010 2nd International Conference on* , vol.1, no., pp.V1-561-V1-563, 5-7, 2010.
55. K. El-Khatib, "Impact of Feature Reduction on the Efficiency of Wireless Intrusion Detection Systems", *IEEE Transactions on Parallel and Distributed Systems, Vol. 21, No. 8,* pp.1143-1149, 2010.
56. S. Shin, T. Kwon, G. Y. Jo, Youngman Park, and HaekyuRhy, "An Experimental Study of Hierarchical Intrusion Detection for Wireless Industrial Sensor Networks", *IEEE Transactions on Industrial Informatics, Vol. 6, No. 4*, pp. 744-757, 2010.
57. X. Yang, G. Zhang, J. Lu, J. Ma, "A Kernel Fuzzy *c*-Means Clustering-Based Fuzzy Support Vector Machine Algorithm for Classification Problems with Outliers or Noises", *IEEE Transactions on Fuzzy Systems, Vol. 19, No. 1,* pp. 105-114, 2011.
58. K.K. Gupta, Baikunthnath, and R. Kotagiri, "Layered Approach Using Conditional Random Fields for Intrusion Detection", *IEEE Transactions on Dependable and Secure Computing*, Vol.7, No.1, pp.35-49, 2010.
59. S. Moustakidis, G. Mallinis, N. Koutsias, J.B. Theocharis, and V. Petridis, "SVM-Based Fuzzy Decision Trees for Classification of High Spatial Resolution Remote Sensing Images", *IEEE Transactions on Geosciences and Remote Sensing, Vol. 50, No.1*, pp.149-169, 2012.

60. C. J. Fung, J. Zhang, I. Aib, and R. Boutaba, "Dirichlet-Based Trust Management for Effective Collaborative Intrusion Detection Networks*", IEEE Transactions on Network and Service Management, Vol. 8, No. 2*, pp.79-91, 2011.

61. S. Umang, B.V.R. Reddy, M.N. Hoda, "Enhanced intrusion detection system for malicious node detection in ad hoc routing protocols using minimal energy consumption", *IET Communications, Vol. 4, No. 17*, pp. 2084–2094, 2010.

62. N. Lu, S. Mabu, and K. Hirasawa, "Integrated Rule Mining Based on Fuzzy GNP and Probabilistic Classification for Intrusion Detection", *Journal of Advanced Computational Intelligence and Intelligent Informatics*, *Vol.15 No.5*, pp.495-496, 2011.

63. J. Liu, F. R. Yu, C.H. Lung, and H. Tang, "Optimal Combined Intrusion Detection and Biometric-Based Continuous Authentication in High Security Mobile Ad Hoc Networks ", IEEE *Transactions on Wireless Communications, Vol. 8, No. 2*, pp.806-815, 2009.

64. Y. Guo, B. Wang, X. Zhao, X. Xie, L. Lin, Q. Zhou, "Feature Selection Based on Rough Set and Modified Genetic Algorithm for Intrusion Detection", *The 5th International Conference on Computer Science & Education Hefei*, China. August 24–27, 2010.

65. G. Song, J. Guo, Y. Nie, "An Intrusion Detection Method based on Multiple Kernel Support Vector Machine", *International Conference on Network Computing and Information Security*, pp.119-123, 2011.

66. G. L. Grinblat, L. C. Uzal, H. A. Ceccatto, and P. M. Granitto, "Solving Non stationary Classification Problems With Coupled Support Vector Machines", *IEEE Transactions on Neural Networks, vol. 22, no. 1*, pp. 37-51, 2011.

67. Lei Li, Jin-Yan Li, Wen-Yan Ding, "A New Method for Colour Image Segmentation Based on FSVM", Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, pp.664-668, 2010.

68. C. Li-ying, Z. Xiao-xian, L. He, and C. Gui-fen, "A Network Intrusion Detection Method Based on Combined Model", *International Conference on Mechatronic Science, Electric Engineering and Computer*, pp.254-257, 2011.

69. K. Cpałka, "A New Method for Design and Reduction of Neuro-Fuzzy Classification Systems", *IEEE Transactions on Neural Networks, Vol. 20, No. 4*, pp. 701-714, 2009.

70. H. Yan, L. Li, F. Di, J. Hua, and Q. Sun, "ANN-based Multi Classifier for Identification of Perimeter Events", *Fourth International Symposium on Computational Intelligence and Design,* pp.158-161, 2011.

71. E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati "Securing XML Documents*", in Proceedings of the International Conference on Extending Database Technology (EDBT2000)*, Konstanz, Germany, pp. 121-135, 2000.

72. R. Bhatti, J.B.D. Joshi, E. Bertino and A. Ghafoor, "Access Control in Dynamic XML-Based Web-Services with XRBAC", *in proceedings of The First International Conference on Web Services,* Las Vegas, pp. 23-26, 2003.

73. X. Huang, H. Wang, Z. Chen and J. Lin, "A Context Rule and Role Based Access Control Model in Enterprise Pervasive Computing Environment", *International Symposium on PC and Applications*, pp. 497 - 502, 2006.

74. F. He and J.J Le "Apply the Technology of RBAC and WS-Security for Secure Web Services Environment in Campus", in Proceedings of the IEEE International Conference on Machine Learning and Cybernetics, pp. 4406 - 4411, 2006.

75. S. R. Sandhu, J. E. Coynek, L. H. Feinsteink, and E. C. Youmank. "Role-Based Access Control Models", *IEEE Computer, Vol. 29, No. 2*, pp.38-47, 1996.

76. N. Li., V. Mahesh and Tripunitara "Security Analysis in Role-Based Access Control", *ACM Transactions on Information and System Security, Vol. 9, No. 4*, pp. 139-420, 2006.

77. S. Barker, "Access Control by Action Control", in Proceedings of the 13th ACM Symposium on Access Control Models and Technologies (SAGMAT), pp. 143-152, 2008.

78. F. Li, W. Wang, Jianfengna, and H. Su. "Action-Based Access Control for Web Services", Journal *of Information Assurance and Security, Vol. 5,* pp. 162-170, 2010.

79. E. Bertino, P. A. Bonatti, and E. Ferrari, "TRBAC: A Temporal Role-Based Access Control Model", *ACM Transactions on Information and System Security, Vol. 4, No. 3*, pp. 191-233, 2001.

80. James B. D. Joshi, E. Bertino, U. Latif, and A. Ghafoor, "A Generalized Temporal Role-Based Access Control Model", *IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 1,* pp. 4 - 23, 2005.

81. L. Chen and J. Crampton, "On Spatio Temporal Constraints and Inheritance in Role Base Access Control", *in Proceedings of the ACM Symposium on Information Computer and Communications Security, ACM,* pp. 205-216, 2008.

82. X. Cui, Y. Chen, and J. Gu "Ex-RBAC: An Extended Role Based Access Control Model for Location-Aware Mobile Collaboration System", *in Proceedings of Second International Conference on Internet Monitoring and Protection,* pp. 36-41, 2007.

83. E. Damiani, D. C. Vimercati, S. Paraboschi, and P. Samarati, "A Fine-Grained Access Control System for XML Documents", *ACM Transactions on Information and System Security, Vol. 5, No. 2*, pp. 169-202, 2002.

84. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models". *IEEE Computer, ISSN 00189162, doi: 10.1109/2.485845, vol. 29, no.2,* pp. 38–47, 1996.

85. L. Chen and J. Crampton, "Inter-domain role mapping and least privilege", *in Proceedings of the 12th ACM symposium on Access control models and technologies (SACMAT '07),* pp. 157-162, 2007.

86. A. Menezes, D. Johnson and S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA*)," Information Security, No.* 1, pp. 36–63, 2001.

87. A, Khalique, K. Singh and S. Sood, "Implementation of Elliptic Curve Digital Signature Algorithm," *International Journal of Computer Applications, vol. 2, No. 2,* pp. 0975 – 8887, 2010.

88. B. Bjurling, R. Steinert, and D. Gillblad, "Translation of probabilistic QoS in hierarchical and decentralized settings". *APNOMS*, pp. 1-8, 2011.

89. Y. Frank, T.L. Chen and Y. Chung Lin, (2010), "An Efficient Date-constraint Hierarchical Key Management Scheme for Mobile Agents," Experts systems with Applications, vol. 37, pp. 7721-7728.

90. J.H. Li, "Hierarchy based Key Assignment Scheme with Date-constraint*," Master Thesis, Feng Chia University*, Taichung, 2004.

91. C. L. Washington and W. Trappe, "Introduction to Cryptography with Coding Theory," *Pearson Education, Second Edition*, pp. 66-70, 2007.

92. P. Vijayakumar, K. Anand, S.Bose, A.Kannan, V.Maheswari, R.Kowsalya, "Hierarchical Key Management Scheme for Securing Mobile Agents with Optimal Computation Time", *to be published in Procedia Engineering, Elsevier*, 2012.

A List of publications

1. K. Anand, S. Ayush, F. Volker, Peter Schoo, and G.Q. Maguire Jr., "N-ary tree based key distribution in a Network as a Service provisioning model", accepted and to be presented at 2012 International Conference on Advances in Computing, Communications and Informatics (ICACCI-2012, 3 – 5 August 2012, Chennai, India (To be published in ACM digital library).

2. K. Anand, G.Q. Maguire Jr., S. Ayush, and Peter Schoo, "Genetic Algorithm based Feature Selection Algorithm for Effective Intrusion Detection in Cloud Networks", to be submitted at KDCloud 2012 "The 3rd International Workshop on Knowledge Discovery Using Cloud and Distributed Computing Platforms", Brussels, Belgium.

3. S. Ayush, K. Anand, F. Volker and I. Schoon, "Bridging security drawbacks of virtualized network resource provisioning model" , EWDCC 2012, Sibu, Romania, presented on May 8th, 2012 , Sibu, Romania (To be published in ACM digital library).

4. P. Vijayakumar, K. Anand, S. Bose, V. Maheswar, R. Kowsalya, and A. Kannan, "Hierarchical key management scheme with optimal computation time", ICMOC – 2012, Chennai, India. April 10-11, 2012, (to be published in Elsevier Procedia).

B IDS publication

The following paper has been submitted to the 3rd International Workshop on Knowledge Discovery Using Cloud and Distributed Computing Platforms (KDCloud, 2012).

K. Anand, G.Q. Maguire Jr., S. Ayush, and Peter Schoo, *"Genetic Algorithm based Feature Selection Algorithm for Effective Intrusion Detection in Cloud Networks"*, to be submitted at KDCloud 2012 “The 3rd International Workshop on Knowledge Discovery Using Cloud and Distributed Computing Platforms”, Brussels, Belgium.

Genetic Algorithm based Feature Selection Algorithm for Effective Intrusion Detection in Cloud Networks

Abstract— Cloud computing is expected to provide on-demand, agile, and elastic services. Cloud networking extends cloud computing by providing virtualized networking functionalities and allows various optimizations, for example to reduce latency while increasing flexibility in the placement, movement, and interconnection of these virtual resources. However, this approach introduces new security challenges. In this paper, we propose a new intrusion detection model in which we combine a newly proposed genetic based feature selection algorithm and an existing Fuzzy Support Vector Machines (SVM) for effective classification as a solution. The feature selection reduces the number of features by removing unimportant features, hence reducing runtime. Moreover, when the Fuzzy SVM classifier is used with the reduced feature set, it improves the detection accuracy. Experimental results of the proposed combination of feature selection and classification model detects anomalies with a low false alarm rate and a high detection rate when tested with the KDD Cup 99 data set.

Keywords- Intrusion Detection System (IDS), Genetic Algorithm (GA), Fuzzy Support Vector Machine (FSVM), tenfold cross validation

I. Introduction

Cloud computing has drastically modified the operating models of organizations reducing both capital and operational expenditures. Moreover, cloud computing virtualizes the physical resources and provides these virtualized resources through provisioning models devised for the cloud ecosystem, such as Software as a service, Platform as a Service, and Infrastructure as a service provisioning models.

Some, cloud service providers provide services to their customers without owning the underlying physical resources, by leasing the physical resources from a cloud vendor, thus reducing their capital and operational expenditures as well as the risks attached with owning and managing the resources. At the same time, the cloud vendor can amortize their already low per-head costs by provisioning their virtualized resource set to more service providers.

The capabilities of cloud computing are enhanced by integrating networking functions with it, which allows the cloud vendor to provision network resources along with other resources. This enables on-demand, dynamic, isolated, and elastic provisioning of virtualized network resources to the end user. In addition, it allows the cloud vendor to optimize various parameters such as network load, network latency, and resource usage. The European Seventh Frame work Program based project Scalable and Adaptive Internet Solutions (SAIL) introduces a new provisioning model, namely Network-as-a-service, which enables the provisioning of virtualized network resources. The project also describes a Cloud Network (CloNe) architecture as the backbone of a service provisioning infrastructure.

The introduction of network resources into the existing cloud computing service provisioning models introduces network-related security challenges. Some of these security challenges were described by Schoo et al., in [12], specifically information security, virtualization environment threats, and communication security. Fusenig et al. in [13] proposed a security architecture for strengthening the CloNe architecture by adding a security goal translation function, identity management function, access control policy function, and an auditing and assurance function. This paper describes the design and deployment of a Genetic Algorithm based Feature Selection Algorithm for effective feature selection as part of an Intrusion Detection System (IDS) using a Fuzzy SVM classifier to support the auditing and assurance function of the CloNe security architecture. Conventional intrusion detection and prevention strategies, firewalls, access control schemes, and cryptographic methods used in the past for providing security to the data communicated through the cloud networks have failed to prove themselves effective for protecting networks and systems from increasingly sophisticated attacks. The proposed IDS turns out to be a suitable solution to these issues. An IDS has become an essential component in security systems since it can be used to detect threats *before* they cause widespread damage. The design and construction of an IDS has many challenges including data collection, data pre-processing, identification of malicious nodes, reporting, and response. Among these activities, identification of malicious nodes is an important and essential activity.

The IDS proposed in this paper can be used to examine collected audit data. Intrusion detection paradigms are based upon models of intrusive or innocent behaviour, so that both internal and external intrusion attempts may be identified efficiently. Moreover, an intelligent IDS should be an effective defensive system that is capable of adapting dynamically to the changing traffic patterns and must be deployed throughout the network, rather than only at servers. This deployment is essential to detect attackers both at the individual nodes and also based on the traffic patterns in the network. The main factor that complicates constructing such an IDS is the necessity for an autonomously evolving system. Complicating this evolution are the huge amounts of network traffic, highly imbalanced data distribution, and the difficulty in recognizing normal versus abnormal behaviour of a user or application. As the audit database grows, even more data has to be considered when forming patterns, otherwise there is increased risk for a high false positive rate. Too high a rate of false positives defeats the purpose of an IDS as either too many alerts are generated that have to be handled manually or there are too many authorized actions which are prevented from occurring. This audit database is frequently augmented with additional real time traffic data. In most IDSs, the IDS is trained on the whole audit database. Our observations reveal that this database contains irrelevant and redundant features impeding the training and testing process, consuming unnecessary resources and leading to poor detection rate.

In order to improve the performance of an IDS, it is necessary to identify and remove the insignificant and duplicate information from the underlying dataset. Therefore, an effective IDS must have a pre-processing component that selects only the necessary features and a classification component for making efficient decisions. These components must work together to optimize the performance with respect to the detection time and to enhance the detection accuracy.

In this paper, we propose an intrusion detection system which uses a new genetic based feature selection algorithm and combine it with a Fuzzy SVM based classifier proposed by Lin [11] in order to create IDS that is effective in identifying intrusions present at the network layer. For this purpose, we propose a new architecture for host based intrusion detection system which will analyse the data present in the network. In order to validate this system, we use the KDD Cup dataset records split in the ratio of 9:1 to provide a tenfold cross validation of the effectiveness of the classifier. The main contribution of the proposed system is that it provides an architectural framework for integrating the feature selection system with the decision making system.

The remainder of this paper is organized as follows: Section 2 surveys related works and compares them with this proposed solution. Section 3 describes a Cloud Network Security Architecture. Section 4 describes the proposed IDS system architecture. Section 5 offers details of the proposed IDS. Section 6 discusses the results obtained with the proposed IDS and a tenfold cross validation performed to test of the system. Section 7 concludes and suggests some possible future enhancements.

II. Related work

There are many works in the literature concerning feature selection and classification. Among them, a heuristic genetic neural network was proposed by Zhang [1] to improve the performance of intrusion detection, in which input features, network structure, and connection weights were considered jointly. Li et al. [2] combined a fuzzy SVM and a multi-class SVM based on a binary tree to develop a network IDS that increases the classification accuracy. Chen et al. [3] discussed the application of fuzzy transitive kernels as fuzzy similarity relations for developing a fuzzy roughest based classifier. They used the lower approximation in fuzzy transitive kernel based fuzzy rough set and assigned memberships to each object. Finally, they used this for classification, thus enhancing the performance of SVM for their application.

Jiang et al. [4] introduced class and sample weighted factors, thus to solve the problem of classification biases caused by uneven training sets. Furthermore, they constructed a decision model based on these samples to improve the classification accuracy. Zaman et al. [5] improved the Support Vector Decision Function approach by integrating it with a fuzzy inferencing model. They used the fuzzy inferencing model to improve the performance of learning approximation used for decision making.

El-Khatib [6] proposed a novel hybrid model that efficiently selects the optimal set of features in order to detect intrusions which are specific to 802.11. Their model for feature selection uses the information gain ratio which is used to compute the relevancy of each feature. Moreover, this system uses the existing k-means classifier to classify and select the required and relevant features of MAC layer which will help to improve the accuracy of intrusion detection systems and at the same time it will reduce learning time of its learning algorithm. Farid et al. [7] enhanced the performance of IDS by proposing a weight based decision tree. Guo et al. [8] proposed a new feature selection algorithm by combining rough sets and genetic algorithm to form a new clustering technique. Stein et al. [9] proposed a decision tree classifier for network intrusion detection with GA-based feature selection. The main advantage of their work is reduction in classification time.

Cao Li-ying et al. [10] combined SVM algorithm and LVQ neural network algorithm and applied it in network intrusion detection systems. They concluded as follows: (1) In contrast with BP neural networks, the convergence speed of SVM-LVQ model is faster and the method is easy to perform. Furthermore, its detection rate of attacks is significantly higher and error rate is lower. (2) The combined model method to obtain the recognition rate has improved significantly than the ordinary method. Krzysztof Cpałka [6] proposed a new class of neuro-fuzzy systems. Moreover, he developed a novel method for reduction of such systems without the deterioration of their accuracy. The reduction algorithm gradually eliminates inputs, rules, antecedents, and the number of discretization points of integrals in the “centre of area” defuzzification method. It then automatically detects and merges similar input and output fuzzy sets. Computer simulations have shown that the accuracy of the system after reduction and merging has not deteriorated, despite the fact that in some cases up to 54% of various parameters and 74% of inputs were eliminated. The reduction algorithm has been tested using well-known classification benchmarks.

The basic requirements of the IDS proposed in this paper are its seamless integration with the overall CloNe (security) architecture and the individual security functions. Moreover, the proposed IDS shall aim to detect the plausible attacks on the CloNe architecture with an acceptable success rate. In this paper, a new intrusion detection system is proposed which differs from existing work in many ways. First, this is a host based IDS and hence is more efficient than the existing network based IDSs. Second, we consider intrusions at the network layer and hence all the important attacks, including Denial of Service attacks, are captured effectively. Third, it uses a pre-processing technique based on genetic algorithms which intelligently performs attribute selection. Fourth, it uses a new classification algorithm for effectively identifying the intruders. Finally, we use tenfold cross validation for validating the decisions made in this system.

III. Cloud Network Security Architecture

Figure 1 describes CloNe architecture, its supporting security functions and their interaction mechanisms. This CloNe security architecture aims to strengthen the CloNe architecture against the security challenges and vulnerabilities described in Schoo et al [12]. Fusenig et al [13] described holistic security architecture to address some of the security challenges of CloNe architecture. The security architecture comprises of a security goal translation function as its backbone, which translates the security requirements given by the different entities in the provisioning infrastructure. The security goal translation translates the security requirements into concrete resource configurations and sends them to the resource administration interface for deployment on the underlying resource set. The security functions include the backbone security goal translation function, auditing and assurance function, access control policy function, and identity management function.

The auditing and assurance function ensures that the security goal translation function and its supporting security functions are functioning in accordance with the operating policies specified by the different participating entities of the provisioning infrastructure. The auditing and assurance function devised for the CloNe architecture builds upon the Cloud Audit specifications, and integrates seamlessly with the supporting functions of the CloNe security architecture. An important addition to the auditing and assurance function can be the integration of an intrusion detection system which provides an acceptable success rate defined and measured using a metric-based approach. The backbone intrusion detection algorithm shall be described in detail in Section 5.

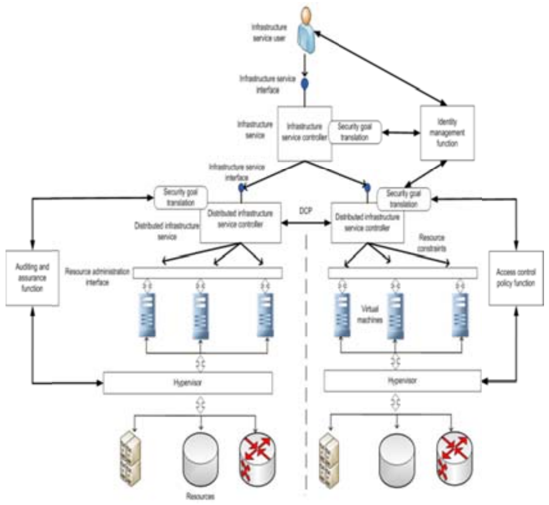

Infrastructure service user
Infrastructure service interface
Infrastructure service
Infrastructure service controller
Security goal translation
Identity management function
Infrastructure service interface
Security goal translation
Distributed infrastructure service controller
DCP
Distributed infrastructure service controller
Security goal translation
Distributed infrastructure service
Resource constraints
Auditing and assurance function
Resource administration interface
Virtual machines
Access control policy function
Hypervisor
Hypervisor
Resources

Figure 1: CloNe security architecture

IV. IDS architecture

The architecture of the IDS proposed in this paper is shown in Figure 2. This IDS consists of four modules: User Interface Module, Feature Selection Module, Classification Module, and Prevention module.

The user interface module collects the networks data from the KDD'99 cup data set. The feature selection module selects the necessary features based on genetic algorithms. The Classification module is used to classify the data by using the Fuzzy SVM [11]. The prevention module decides whether the decision made by the classification module on the first set of records is valid and prevents the attacks.

V. Details of the Proposed IDS

In this paper, we propose a new intrusion detection system in which we have developed a new genetic algorithm based feature selection algorithm. Moreover, an effective classification algorithm called Fuzzy Support Vector Machines [11] has been used in this work for effective classification of network trace data. From the approaches used in this work, it was possible to find the intruders effectively.

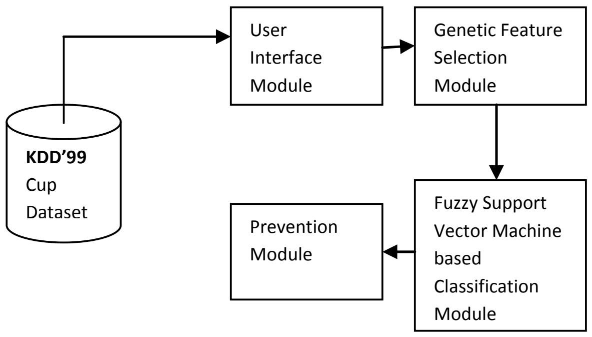

User Interface Module
Genetic Feature Selection Module
KDD'99 Cup Dataset
Prevention Module
Fuzzy Support Vector Machine based Classification Module

Figure 2: Proposed IDS architecture

Proposed Feature Selection Technique Using GA

Genetic based feature selection algorithm has been used in this work in order to select suitable subset of features so that they are potentially useful in classification. Another advantage of GA based feature selection in this work is that it finds and eliminates the redundant features if any because these redundant features may misguide in clustering or classification. The reduction in number of features reduces the training time and ambiguousness, thus a weighted sum genetic feature selection algorithm has been proposed which has increased global search capability and is better in attribute interaction when compared to other algorithms such as the greedy method.

Proposed Framework for Genetic Feature Selection

Subset generation use a method of heuristic search by Lasarczyk et al. [14], in which each instance in the search space specifies a candidate solution for subset evaluation. The decision process is determined by some basic issues. Initially, the search starting point is decided since it controls the direction of search. Feature selection search starts either with a null set with features added one by one or it starts with a full set of features and eliminates them one by one. Both these methods have the drawback of being trapped into local optima. In order to avoid this we employ a random search. Figure 3 shows the framework for our proposed genetic based feature selection approach. This framework uses genetic operations to identify & select the most relevant features of the 41 features present in the KDD cup data set.

Next, a search strategy is decided. A dataset with N features has 2N candidate subsets. This value is very large for moderate and large value of N. In our test case, there are 41 candidate subsets which are quite large. There are three different types of search strategies. They are complete, sequential, and random. Complete searches like branch and bound are exhaustive searches. Sequential search such as greedy hill climbing add or remove features one at a time and find optimal features.

Random search generates the subset in a completely random manner, i.e., it does not follow any deterministic rule. When compared to above two approaches, the utilization of randomness helps to avoid the local optima problem occurring in the heuristic search space to obtain an optimal subset.

Evaluation of Subset

After the subset is generated, it is evaluated using an evaluation criterion. The best or optimal subset of features obtained using one criterion may not be optimal according to another criterion. Based on the dependency of evaluation of a subset using the classification or clustering algorithm applied at the end, feature subset evaluation criterion can be classified into independent or dependent criterion. Commonly used independent criteria are distance, information, dependency, and consistency measures. If a feature incurs greater difference computed using the above criteria than other features, then the feature that incurs greater difference is considered. This evaluation criterion uses the intrinsic characteristics of the dataset without applying any classification or clustering algorithms.

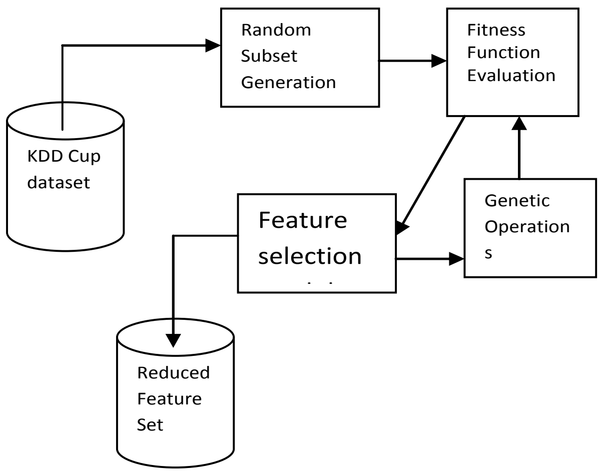


Figure 3: Feature Selection Systems

On the other hand, dependent criterion uses the performance of the classification or clustering algorithm on the selected feature subset in identifying essential features. This approach gives superior performance as it selects features based on the classification or clustering algorithm applied. The approach proposed uses dependent criterion for selecting significant features which are to be used in the detection process. Here predictive accuracy and feature count are used as the primary measures. Even though the computational complexity of this approach is higher than for an independent measure, it provides greater detection accuracy. Since feature selection is performed offline, the complexity involved in this is not related to the detection process and hence the time taken is immaterial.

Stopping Criteria

A stopping criterion determines when the feature extraction algorithm should stop. The proposed algorithm terminates, when any one of the following condition is met:

(i.) The search completes when the maximum number of iteration is reached or (ii.) When a good subset is selected i.e., the difference between previous fitness and current fitness is less than the given tolerance value.

Validation of Results

One direct way of result validation is based on the prior knowledge about the data. However, in real-world applications, such prior knowledge is not available. Hence, the proposed approach relies on an indirect method which monitors the change of the detection algorithm performance with a change of features. Experiments have been conducted with the full set of features and selected subset of features to compare the performance of classifier. From these experiments, it has been found that the detection accuracy is almost the same in both the cases. Therefore, feature selection can be carried out to improve the performance of the system. In addition, tenfold cross validation was performed to ensure the correctness of classification accuracy.

Proposed Genetic based Feature Selection Algorithm

Algorithm: Feature set selection using weighted sum GA.

Input: Network traffic pattern (All features), Number of generations, Population size, Crossover probability (Pc), Mutation probability (Pm).

Output: Set of selected features.

```
Genetic_Feature_Selection ( ) {
1. Initialize the population randomly with the size of each chromosome equal to the total number of features in the dataset which is equal to 41. Each gene value in the chromosome can be '0' or '1'. A bit value of '0' represents that the corresponding feature is not present in chromosome and '1' represents that the feature is present.
2. Initialize the weights W1 = 0.7, W2 = 0.3, N (total number of records in the training set), Pc and Pm.
3. For each chromosome in the new population {
a. Apply uniform crossover with a probability Pc.
b. Apply mutation operator to the chromosome with a probability Pm.
c. Evaluate fitness = W1 * Accuracy + W2 * (1/ Count of   Ones)}
4. If (Current fitness – Previous fitness < 0.0001) then exit
5. Select the top best 60% of chromosomes into new population using tournament selection.
6. If number of generations is not reached, go to line 3.
}
```

Experiment Test Bed

The simple test bed consisted of two computers with similar hardware (CPU: Intel Pentium i7-2720QM (6 MB cache, 2.20 GHz), RAM: 8 GB, NIC: Intel Ultimate-N 6300 (802.11 a/b/g/n), Hard Drive: Seagate 500GB) connected with a cross over cable. Each physical computer hosted five virtual machines (VMs) with Proxmox VE 1.8, an open source virtualization environment. Each VM was running a copy of Microsoft's Windows 7 operating system. We used IPv4 as the communication protocol stack in both the VMs and underlying operating system. Proxmox VE uses a bridged networking model. These bridges are similar to physical network switches, but implemented in software on the underlying Proxmox VE host. All VMs share a single bridge, thus it was as if virtual network cables from each guest were all plugged into a single physical switch. To avoid cross VM communication, VLANs (implementing IEEE 802.1q) are used to separate the networks as if each VM were separately connected to the underlying physical system. Each VM can act as a master or a slave depending on the deployed application. The master node can use resources of one or more slaves at any given time. Each VM has to authenticate every other VM before sharing resources. Even if a VM belongs to the same logical rack, they authenticate each other and communications are routed through the virtual router whose routing daemon is running on the underlying physical machine.

VI. Results and Discussion

In this work, a new genetic based feature selection approach has been proposed and implemented in order to select subset of important features from the original feature set of KDD cup data set. The important features selected by the genetic based feature selection algorithm are used for classifying the data set in order to find the intrusions. In classification, the existing fuzzy SVM is used in this work. By this process, the 17 relevant features shown in Table 2 have been generated by computing the weighted sum GA values from the 41 features shown in Table 1, using the proposed feature selection algorithm. From the experiments conducted using these features, it has been observed that feature selection reduces the training and testing time and at the same time produces similar accuracy as that of full feature set.

Table 1: Features of KDD Cup dataset

S.No	Feature Name	S. No	Feature Name
1	duration	22	is_guest_login
2	protocol_type	23	Count
3	service	24	serror_rate
4	src_byte	25	rerror_rate
5	dst_byte	26	same_srv_rate
6	flag	27	diff_srv_rate
7	land	28	srv_count
8	wrong_fragment	29	srv_serror_rate
9	urgent	30	srv_rerror_rate
10	hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	num_compromised	34	dst_host_same_srv_count
14	root_shell	35	dst_host_diff_srv_count
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	dst_host_serror_rate
18	num_shells	39	dst_host_srv serror rate
19	num_access_shells	40	dst_host_rerror_rate
20	num_outbound_cmds	41	dst_host_srv_rerror_rate
21	is_hot_login		

Table 2: 2 Name of the 17 selected features

Name of the selected features
protocol_type, service, flag, src_bytes, dst_bytes, wrong_fragment, hot, logged_in, count, serror_rate, same_srv_rate, diff_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_rerror_rate

Table 3 shows the comparison of detection rates between SVM and Fuzzy SVM when they are applied with 17 features returned by the feature selection algorithm. From this table, it can be observed that the detection accuracy of fuzzy SVM is higher than that of neural networks and SVM. Therefore, fuzzy SVM has been selected the proposed IDS intrusion detection system. Table 4 provides the comparative analysis of error rates between SVM and Fuzzy SVM classifiers when they are applied with the 17 features returned by the feature selection algorithm. From this table, it has been observed that the error rate is less in the fuzzy SVM when it is compared with neural network and SVM.

Table 5 shows the comparison of the classifier SVM and Fuzzy SVM with full features and also with the fuzzy SVM classification with 17 features. From this table, it can be observed that the error rate is reduced in fuzzy SVM leading to increase in classification accuracy when it is compared with the other two methods. Moreover, the detection time is also reduced when fuzzy SVM is applied with the reduced number of features. The increase in classification accuracy is due to the fact that the confusions made by irrelevant attributes are eliminated by feature reduction. In addition, the classification time is reduced due to the reduction in the number of rules to be applied for decision making with reduced number of attributes.

Table 3: Detection Rate Comparisons of SVM and Fuzzy SVM Approaches with feature selection

Class Types	Total Test Samples	Detection Rate (%)	
		SVM	Fuzzy SVM
Normal	47911	96.03	98.75
DoS	7458	90.6	98.3
R2L	2754	53.74	85.48
U2R	200	83.5	89
Probe	2421	83.97	96.53

Table 4: Error Rate Comparisons of NN, SVM and Fuzzy SVM Approaches

Class Types	Total Test Samples	Error Rate (%)	
		SVM	Fuzzy SVM
Normal	47911	3.98	1.25
DoS	7458	5.41	2.70
R2L	2754	4.62	1.45
U2R	200	1.65	1.10
Probe	2421	1.60	1.47

Table 5: Comparison of SVM, Fuzzy SVM and Feature selected Fuzzy SVM Methods

Algorithm	**Detection Accuracy (%)**	**Error Rate (%)**	**Training Time (Milli Sec)**
SVM	83.5821	6.5147	846
FSVM	92.4612	5.2136	223
Feature Selection +FSVM	98.5123	3.134	112

Figure 4 shows the performance analysis for the four types of attacks namely DoS, Probe, User to Root (U2R) and Remote to Local (R2L) when they are detected with fuzzy SVM with 41 features and fuzzy SVM with 17 features. From Figure 4, it can be seen that the detection accuracy is improved when the fuzzy SVM is applied with 17 features. This is due to the fact that the fuzzy rules applied to 17 features are less in number and do not conflicting with each other during decision making.

Figure 5 depicts the false alarm rate produced by fuzzy SVM with 41 features and fuzzy SVM with 17 features in five experiments namely E1, E2, E3, E4 and E5. From the figure, it can be observed that the false alarm rate is reduced when the fuzzy SVM is applied with 17 features. This is due to the fact that the decision accuracy is greater in the feature selected FSVM.

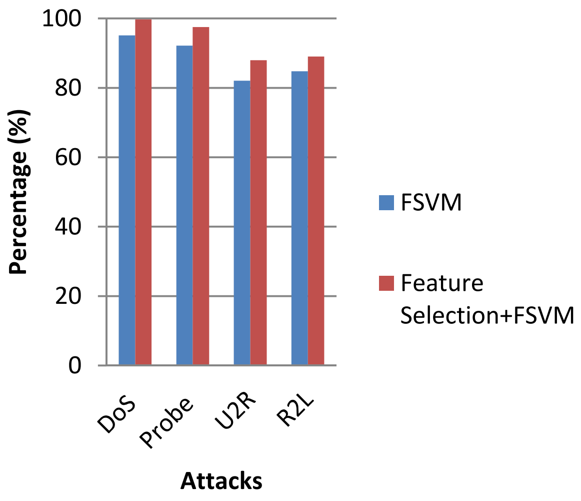


Figure 4: Performance analyses for attacks

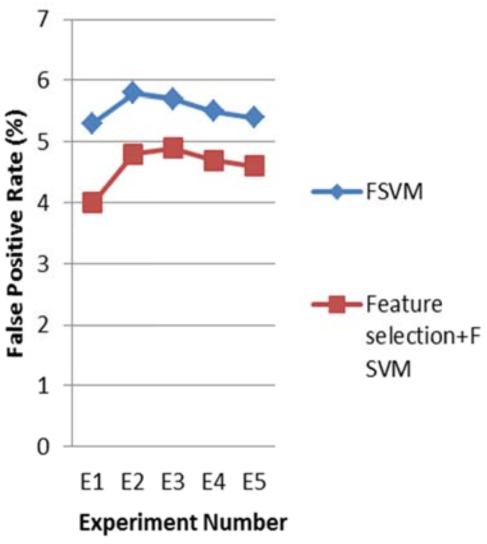


Figure 5: False alarm rate comparisons

Conclusions and Future Enhancements

In this paper, a new genetic based feature selection algorithm for cloud networks has been proposed. This algorithm is used to select optimal number of features from the KDD cup data set for intrusion detection. Moreover, a framework for intrusion detection that uses this feature selection algorithm and then applies the existing Fuzzy SVM for effective classification of intrusions using KDD Cup dataset for securing the cloud networks has been proposed. The main advantage of the proposed genetic algorithm based feature selection algorithm is that it improves the detection accuracy of the fuzzy SVM classifier by providing minimum and required number of features. This helps to reduce the classification time in addition to the increase in classification accuracy. Future work in this direction would extend the fuzzy SVM with rough sets to further improve the detection accuracy.

Acknowledgment

The authors would like to express their gratitude to the European Commission for its funding through the “Scalable and Adaptive Internet Solutions”, SAIL Project (FP7-ICT-2009-5-257448).

References

1. Biying Zhang, “A Heuristic Genetic Neural Network for Intrusion Detection”, 2011 International Conference on Internet Computing and Information Services, pp. 510-511, 2011.
2. Lei Li, Zhi-ping Gao, and Wen-yan Ding, ‘Fuzzy Multi-class Support Vector Machine Based on Binary Tree in Network Intrusion Detection’, in Proceedings of the 2010 International Conference on Electrical and Control Engineering, Washington, DC, USA, 2010, pp. 1043–1046, DOI:10.1109/iCECE.2010.264.
3. Degang Chen, Qiang He, and Xizhao Wang, ‘FRSVMs: Fuzzy rough set based support vector machines’, Fuzzy Sets Syst., vol. 161, no. 4, pp. 596–607, February 2010, DOI:10.1016/j.fss.2009.04.007.
4. Jiaqi Jiang, Ru Li, Tianhong Zheng, Feiqin Su, and Haicheng Li, “A new intrusion detection system using Class and Sample Weighted C-Support Vector Machine, 2011 Third International Conference on Communications and Mobile Computing, pp.51-54, 2011.

5. Safaa Zaman and Fakhri Karray, “Fuzzy ESVDF approach for Intrusion Detection Systems”,International conference on Advanced Information Networking and Applications pp.539-545, 2009.

6. Khalil El-Khatib, “Impact of Feature Reduction on the Efficiency of Wireless Intrusion Detection Systems” IEEE Transactions on Parallel and Distributed Systems, Vol. 21, No. 8, pp. 1143-1149, 2010.

7. Dewan Md. Farid, Nouria Harbi, Emna Bahri, Mohammad Zahidur Rahman, and Chowdhury Mofizur Rahman, “Attacks Classification in Adaptive Intrusion Detection using Decision Tree”, World Academy of Science, Engineering and Technology, Vol. 63, pp. 86-90, 2010.

8. Yuteng Guo,Beizhan Wang, Xinxing Zhao, Xiaobiao Xie, Lida Lin, and Qingda Zhou, “Feature Selection Based on Rough Set and Modified Genetic Algorithm for Intrusion Detection”, The 5th International Conference on Computer Science & Education, pp. 1441-1446, 2010.

9. Gary Stein, Bing Chen, Annie S. Wu, and Kien A. Hua, ‘Decision tree classifier for network intrusion detection with GA-based feature selection’, in Proceedings of the 43rd annual Southeast regional conference - Volume 2, New York, NY, USA, 2005, pp. 136–141, DOI:10.1145/1167253.1167288.

10. Cao Li-ying, Zhang Xiao-xian, Liu He, and Chen Gui-fen, “A Network Intrusion Detection Method Based on Combined Model”, International Conference on Mechatronic Science, Electric Engineering and Computer, pp.254-257, 2011.

11. Chun-Fu Lin and Shen-De Wang, “Fuzzy Support Vector Machines”, IEEE Transactions on Neural Networks, Vol. 13, No. 2, pp. 464-471, 2002.

12. P. Schoo, V. Fusenig, V. Souza, M.Melo, P. Murray, H. Debar, H. Medhioub, and D. Zeghlache, “Challenges for cloud networking security,” in Mobile Networks and Management, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer, Heidelberg, 2010.

13. Volker Fusenig and Ayush Sharma. Security architecture for cloud networking. Presented in Proceedings of the 2012 International Conference on Networking and Computing, ICNC 2012, IEEE Computer Society, 2012.

14. Christian W.G. Lasarczyk, Peter Dittrich, and Wolfgang Banzhaf, “Dynamic Subset Selection Based on a Fitness Case Topology”, Evolutionary Computation, Vol. 12, No. 2 , pp. 223-242, 2004.

TRITA-ICT-EX-2012:174