# CLOUD SECURITY USING ELLIPTIC CURVE CRYPTOGRAPHY AND DIFFIE HELLMAN

**A Thesis Submitted**
**in Partial Fulfillment of the Requirements**
**For the Degree of**

## MASTER OF TECHNOLOGY

**in**
**COMPUTER SCIENCE & ENGINEERING**
**(Specialization: Software Engineering)**

**by**

**Rishi Kumar**
**Enrollment no. 11704490642**

**Under the Supervision of**
**Assistant Professor Abhinav Singh**

**to the**

**SCHOOL OF ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE**

**BABU BANARASI DAS UNIVERSITY**
**LUCKNOW**

**May, 2019**

# CERTIFICATE

It is certified that the work contained in this thesis entitled **“CLOUD SECURITY USING ELLIPTIC CURVE CRYPTOGRAPHY AND DIFFIE HELLMAN”,** by **Rishi Kumar** (Roll No: 1170449006), for the award of **Master of Technology** from Babu Banarasi Das University has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree.

Signature
Abhinav Singh
Assistant professor
Babu Banarasi DasUniversity Lucknow
Date:

# ABSTRACT

In cryptography, key exchange is a strategy by which cryptographic keys are exchanged between two gatherings and those keys are utilized as a part of some cryptographic algorithms like AES. Utilizing those keys sender and recipient exchange encrypted messages. Public key cryptography gives a secured strategy to exchange secret keys. The key exchange issue is the means by which gatherings exchange the keys or data in a communication channel so that nobody else other than sender and recipient can get those. This paper presents Diffie-Hellman key exchange, a procedure which is one of the public key crypto- graphic protocols used to build up a secret key between two gatherings over a frail channel. The protocol itself is constrained to exchange of the keys i.e, we are not sharing data while the key exchange, and we are making a key together. We start with implementation of algorithm i.e., by building up a mutual secret between two gatherings that can be utilized for secret communication for exchanging information over a public channel. Having no entity authentication mechanism, protocol is electively assaulted by the man-in-the-middle attack and impersonation attack in practically speaking. Diffie-Hellman is appropriate for utilization in information communication however is less frequently utilized for information storage or archived over long period of time.

## ACKNOWLEDGMENTS

I take this opportunity to express a deep sense of gratitude towards my guide **Mr. Abhinav Singh,** for providing excellent guidance, encouragement, support and inspiration throughout the project work. Without her invaluable guidance, this work would never have been a success. I would also like to thank all my classmates for their valuable suggestions and helpful discussion specially in making report.

**Rishi Kumar**

Babu Banarasi Das University Lucknow
May, 2019

# TABLE OF CONTENT

Page No.

# List of Tables

# List of Figures

# List of Abbreviations

| | |
|---|---|
| ECC | Elliptic Curve Cryptography |
| DH | Diffie–Hellman |
| SaaS | Software as a service |
| PaaS | Platform as a service |
| Iaas | Infrastructure as service |
| CSP | Cloud Service Provider |
| DSA | Digital Signature Algorithm |
| RSA | Rivest Shamir Adleman |
| AES | Advanced Encryption Standard |

# CHAPTER 1

## 1.1 Introduction

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. The term is generally used to describe data centers available to many users over the Internet. Large clouds, predominant today, often have functions distributed over multiple locations from central servers. If the connection to the user is relatively close, it may be designated an edge server.

Clouds may be limited to a single organization (enterprise clouds,) be available to many organizations (public cloud,) or a combination of both (hybrid cloud). The largest public cloud is Amazon AWS.

Cloud computing relies on sharing of resources to achieve coherence and economies of scale.

Advocates of public and hybrid clouds note that cloud computing allow companies to avoid or minimize up-front IT infrastructure costs. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and that it enables IT teams to more rapidly adjust resources to meet fluctuating and unpredictable demand. Cloud providers typically use a "pay-as-you-go" model, which can lead to unexpected operating expenses if administrators are not familiarized with cloud-pricing models.

The availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing has led to growth in cloud computing

## 1.2 History

While the term "cloud computing" was popularized with Amazon.com releasing its Elastic Compute Cloud product in 2006, references to the phrase "cloud computing" appeared as early as 1996, with the first known mention in a Compaq internal document.

The cloud symbol was used to represent networks of computing equipment in the original ARPANET by as early as 1977, and the CSNET by 1981 — both predecessors to the Internet itself. The word cloud was used as a metaphor for the Internet and a standardized cloud-like shape was used to denote a network on telephony schematics. With this simplification, the implication is that the specifics of how the end points of a network are connected are not relevant for the purposes of understanding the diagram.

The term cloud was used to refer to platforms for distributed computing as early as 1993, when Apple spin-off General Magic and AT&T used it in describing their (paired) Typescript and Personal ink technologies. In Wired's April 1994 feature "Bill and Andy's Excellent Adventure II", Andy Herzfeld commented on Typescript, General Magic's distributed programming language:

"The beauty of Typescript ... is that now, instead of just having a device to program, we now have the entire Cloud out there, where a single program can go and travel to many different sources of information and create sort of a virtual service. No one had conceived that before. The example Jim White [the designer of Typescript, X.400and ASN.1] uses now is a date-arranging service where a software agent goes to the flower store and orders flowers and then goes to the ticket shop and gets the tickets for the show, and everything is communicated to both parties Since 2000, cloud computing has come into existence.

In April 2008, Google released Google App Engine in beta.

In early 2008, NASA's Open Nebula

, enhanced in the RESERVOIR European Commission-funded project, became the first open-source software for deploying private and hybrid clouds, and for the federation of clouds.

By mid-2008, Gartner saw an opportunity for cloud computing "to shape the relationship among consumers of IT services, those who use IT services and those who sell them “and observed that "organizations are switching from company-owned hardware and software assets to per-use service-based models" so that the "projected shift to computing ... will result in dramatic growth in IT products in some areas and significant reductions in other areas."

In 2008, the U.S. National Science Foundation began the Cluster Exploratory program to fund academic research using Google-IBM cluster technology to analyze massive amounts of data,

In February 2010, Microsoft released Microsoft Azure, which was announced in October 2008.

In July 2010, Rack space Hosting and NASA jointly launched an open-source cloud-software initiative known as Open Stack. The Open Stack project intended to help organizations offering cloud-computing services running on standard hardware. The early code came from NASA's Nebula platform as well as from Rack space's Cloud Files platform. As an open source offering and along with other open-source solutions such as Cloud Stack, Gamete and Open Nebula, it has attracted attention by several key communities. Several studies aim at comparing these open sources offerings based on a set of criteria.

On March 1, 2011, IBM announced the IBM Smart Cloud framework to support Smarter Planet. Among the various components of the Smarter Computing foundation, cloud computing is a critical part. On June 7, 2012, Oracle announced the Oracle Cloud. This cloud offering is poised to be the first to provide users with access to an integrated set of IT solutions, including the Applications (SaaS), Platform (PaaS), and Infrastructure (IaaS) layers.

In May 2012, Google Compute Engine was released in preview, before being rolled out into General Availability in December 2013.

Cloud computing exhibits the following key characteristics:

Agility for organizations may be improved, as cloud computing may increase users' flexibility with re-provisioning, adding, or expanding technological infrastructure resources.

Cost reductions are claimed by cloud providers. A public-cloud delivery model converts capital expenditures (e.g., buying servers) to operational expenditure. This purportedly lowers barriers to entry, as infrastructure is typically provided by a third party and need not be purchased for one-time or infrequent intensive computing tasks. Pricing on a utility computing basis is "fine-grained", with usage-based billing options. As well, less in-house IT skills are required for implementation of projects that use cloud computing. The e-FISCAL project's state-of-the-art repository contains several articles looking into cost aspects in more detail, most of them concluding that costs savings depend on the type of activities supported and the type of infrastructure available in-house.

Device and location independence enable users to access systems using a web browser regardless of their location or what device they use (e.g., PC, mobile phone). As infrastructure is off-site (typically provided by a third-party) and accessed via the Internet, users can connect to it from anywhere.

Maintenance of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places (e.g., different work locations, while travelling, etc.).

Multitenancy enables sharing of resources and costs across a large pool of users thus allowing for:

centralization of infrastructure in locations with lower costs (such as real estate, electricity, etc.)

peak-load capacity increases (users need not engineer and pay for the resources and equipment to meet their highest possible load-levels)

utilization and efficiency improvements for systems that are often only 10–20% utilized.

Performance is monitored by IT experts from the service provider, and consistent and loosely coupled architectures are constructed using web services as the system interface.

Productivity may be increased when multiple users can work on the same data simultaneously, rather than waiting for it to be saved and emailed. Time may be saved as information does not need to be re-entered when fields are matched, nor do users need to install application software upgrades to their computer.

Reliability improves with the use of multiple redundant sites, which makes well-designed cloud computing suitable for business continuity and disaster recovery.

Scalability and elasticity via dynamic ("on-demand") provisioning of resources on a fine-grained, self-service basis in near real-time (Note, the VM startup time varies by VM type, location, OS and cloud providers), without users having to engineer for peak loads. This gives the ability to scale up when the usage need increases or down if resources are not being used. Emerging approaches for managing elasticity include the utilization of machine learning techniques to propose efficient elasticity models.

Security can improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because service providers are able to devote resources to solving security issues that many customers cannot afford to tackle or which they lack the technical skills to address. However, the complexity of security is greatly increased when data is distributed over a wider area or over a greater number of devices, as well as in multi-tenant systems shared by unrelated users. In addition, user access to security audit logs may be difficult or impossible. Private cloud Installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.

## 1.3 Service models

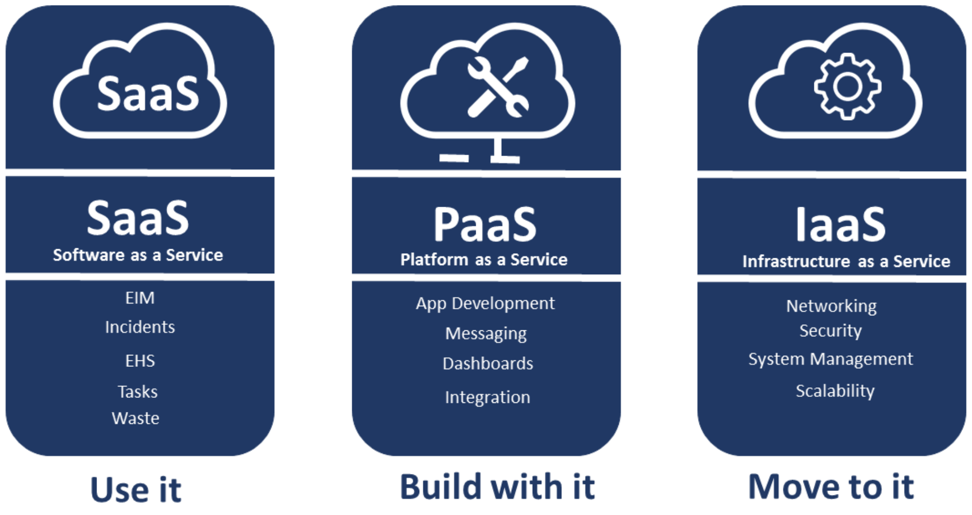


Fig: 1.3

Cloud computing service models arranged as layers in a stack

Though service-oriented architecture advocates "everything as a service" (with the acronyms EaaS or XaaS, or simply aas), cloud-computing providers offer their "services" according to different models, of which the three standard models per NIST are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). These models offer increasing abstraction; they are thus often portrayed as a layers in a stack: infrastructure-, platform- and software-as-a-service, but these need not be related. For example, one can provide SaaS implemented on physical machines (bare metal), without using underlying PaaS or IaaS layers, and conversely one can run a program on IaaS and access it directly, without wrapping it as SaaS.

### 1.3.1 Infrastructure as a service (IaaS)

"Infrastructure as a service" (IaaS) refers to online services that provide high-level APIs used to dereference various low-level details of underlying network infrastructure like physical computing resources, location, data partitioning, scaling, security, backup etc. A hypervisor runs the virtual machines as guests. Pools of hypervisors within the cloud operational system can support large numbers of virtual machines and the ability to scale services up and down according to customers' varying requirements. Linux containers run in isolated partitions of a single Linux kernel running directly on the physical hardware. Linux cgroups and namespaces are the underlying Linux kernel technologies used to isolate, secure and manage the containers. Containerization offers higher performance than virtualization, because there is no hypervisor overhead. Also, container capacity auto-scales dynamically with computing load, which eliminates the problem of over-provisioning and enables usage-based billing. IaaS clouds often offer additional resources such as a virtual-machine disk-image library, raw block storage, file or object storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles.

The NIST's definition of cloud computing describes IaaS as "where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

IaaS-cloud providers supply these resources on-demand from their large pools of equipment installed in data centers. For wide-area connectivity, customers can use either the Internet or carrier clouds (dedicated virtual private networks). To deploy their applications, cloud users install operating-system images and their application software on the cloud infrastructure. In this model, the cloud user patches and maintains the operating systems and the application software. Cloud providers typically bill IaaS services on a utility computing basis: cost reflects the amount of resources allocated and consumed.

### 1.3.2 Platform as a service (PaaS)

The NIST's definition of cloud computing defines Platform as a Service as:

The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

PaaS vendors offer a development environment to application developers. The provider typically develops toolkit and standards for development and channels for distribution and payment. In the PaaS models, cloud providers deliver a computing platform, typically including operating system, programming-language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. With some PaaS offers like Microsoft Azure, Oracle Cloud Platform and Google App Engine, the underlying computer and storage resources scale automatically to match application demand so that the cloud user does not have to allocate resources manually. The latter has also been proposed by an architecture aiming to facilitate real-time in cloud environments.

Some integration and data management providers have also embraced specialized applications of PaaS as delivery models for data solutions. Examples include iPaaS (Integration Platform as a Service) and dPaaS (Data Platform as a Service). iPaaS enables customers to develop, execute and govern integration flows. Under the iPaaS integration model, customers drive the development and deployment of integrations without installing or managing any hardware or middleware. dPaaS delivers integration—and data-management—products as a fully managed service. Under the dPaaS model, the PaaS provider, not the customer, manages the development and execution of data solutions by building tailored data applications for the customer. PaaS users retain transparency and control over data through data-visualization tools. Platform as a Service (PaaS) consumers do not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but have control over the deployed applications and possibly configuration settings for the application-hosting environment.

### 1.3.3 Software as a service (SaaS)

The NIST's definition of cloud computing defines Software as a Service as:

The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

In the software as a service (SaaS) model, users gain access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. SaaS is sometimes referred to as "on-demand software" and is usually priced on a pay-per-use basis or using a subscription fee. In the SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. Cloud users do not manage the cloud infrastructure and platform where the application runs. This eliminates the need to install and run the application on the cloud user's own computers, which simplifies maintenance and support. Cloud applications differ from other applications in their scalability—which can be achieved by cloning tasks onto multiple virtual machines at run-time to meet changing work demand. Load balancers distribute the work over the set of virtual machines. This process is transparent to the cloud user, who sees only a single access-point. To accommodate a large number of cloud users, cloud applications can be multitenant, meaning that any machine may serve more than one cloud-user organization.

The pricing model for SaaS applications is typically a monthly or yearly flat fee per user, so prices become scalable and adjustable if users are added or removed at any point. Proponents claim that SaaS gives a business the potential to reduce IT operational costs by outsourcing hardware and software maintenance and support to the cloud provider. This enables the business to reallocate IT operations costs away from hardware/software spending and from personnel expenses, towards meeting other goals. In addition, with applications hosted centrally, updates can be released without the need for users to install new software. One drawback of SaaS comes with storing the users' data on the cloud provider's server. As a result,[citation needed] there could be unauthorized access to the data.

## 1.4 Architecture

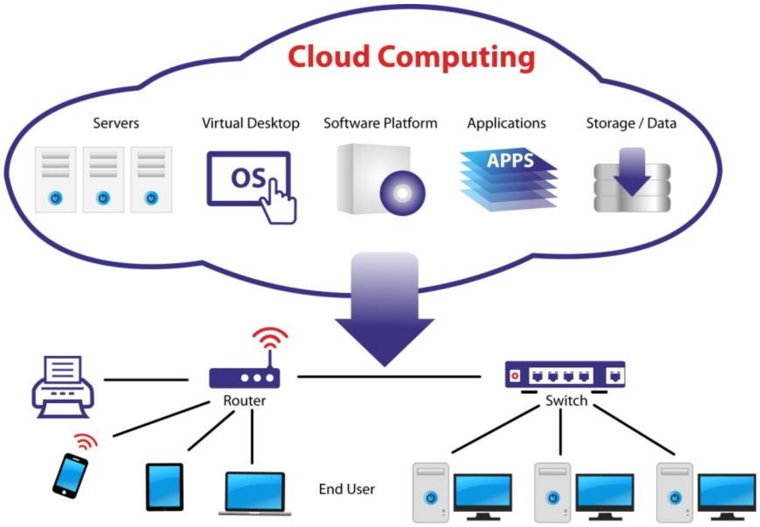


Fig: 1.4 Cloud system architecture

Cloud architecture, the systems architecture of the software systems involved in the delivery of cloud computing, typically involves multiple cloud components communicating with each other over a loose coupling mechanism such as a messaging queue. Elastic provision implies intelligence in the use of tight or loose coupling as applied to mechanisms such as these and others.

Cloud engineering is the application of engineering disciplines to cloud computing. It brings a systematic approach to the high-level concerns of commercialization, standardization, and governance in conceiving, developing, operating and maintaining cloud computing systems. It is a multidisciplinary method encompassing contributions from diverse areas such as systems, software, web, performance, information technology engineering, security, platform, risk, and quality engineering.

### 1.5 Security and Privacy

Cloud computing poses privacy concerns because the service provider can access the data that is in the cloud at any time. It could accidentally or deliberately alter or delete information. Many cloud providers can share information with third parties if necessary for purposes of law and order without a warrant. That is permitted in their privacy policies, which users must agree to before they start using cloud services. Solutions to privacy include policy and legislation as well as end users' choices for how data is stored. Users can encrypt data that is processed or stored within the cloud to prevent unauthorized access. Identity management systems can also provide practical solutions to privacy concerns in cloud computing. These systems distinguish between authorized and unauthorized users and determine the amount of data that is accessible to each entity The systems work by creating and describing identities, recording activities, and getting rid of unused identities.

According to the Cloud Security Alliance, the top three threats in the cloud are Insecure Interfaces and API's, Data Loss & Leakage, and Hardware Failure—which accounted for 29%, 25% and 10% of all cloud security outages respectively. Together, these forms shared technology vulnerabilities. In a cloud provider platform being shared by different users there may be a possibility that information belonging to different customers resides on same data server. Additionally, Eugene Schultz, chief technology officer at Imagined Security, said that hackers are spending substantial time and effort looking for ways to penetrate the cloud. "There are some real Achilles' heels in the cloud infrastructure that are making big holes for the bad guys to get into". Because data from hundreds or thousands of companies can be stored on large cloud servers, hackers can theoretically gain control of huge stores of information through a single attack—a process he called "hyper jacking". Some examples of this include the Dropbox security breach, and iCloud 2014 leak. Dropbox had been breached in October 2014, having over 7 million of its users passwords stolen by hackers in an effort to get monetary value from it by Bitcoins (BTC). By having these passwords, they are able to read private data as well as have this data be indexed by search engines (making the information public).

There is the problem of legal ownership of the data (If a user stores some data in the cloud, can the cloud provider profit from it?). Many Terms of Service agreements are silent on the question of ownership. Physical control of the computer equipment (private cloud) is more secure than having the equipment off site and under someone else's control (public cloud). This delivers great incentive to public cloud computing service providers to prioritize building and maintaining strong management of secure services. Some small businesses that don't have expertise in IT security could find that it's more secure for them to use a public cloud. There is the risk that end users do not understand the issues involved when signing on to a cloud service (persons sometimes don't read the many pages of the terms of service agreement, and just click "Accept" without reading). This is important now that cloud computing is becoming popular and required for some services to work, for example for an intelligent personal assistant (Apple's Siri or Google Now). Fundamentally, private cloud is seen as more secure with higher levels of control for the owner, however public cloud is seen to be more flexible and requires less time and money investment from the user.

## 1.6 Limitations and Disadvantages

According to Bruce Schneider, "The downside is that you will have limited customization options. Cloud computing is cheaper because of economics of scale, and — like any outsourced task — you tend to get what you get. A restaurant with a limited menu is cheaper than a personal chef who can cook anything you want. Fewer options at a much cheaper price: it's a feature, not a bug." He also suggests that "the cloud provider might not meet your legal needs" and that businesses need to weigh the benefits of cloud computing against the risks. In cloud computing, the control of the back end infrastructure is limited to the cloud vendor only. Cloud providers often decide on the management policies, which moderates what the cloud users are able to do with their deployment. Cloud users are also limited to the control and management of their applications, data and services. This includes data caps, which are placed on cloud users by the cloud vendor allocating certain amount of bandwidth for each customer and are often shared among other cloud users.

Privacy and confidentiality are big concerns in some activities. For instance, sworn translators working under the stipulations of an NDA, might face problems regarding data that are not encrypted.

Cloud computing is beneficial to many enterprises; it lowers costs and allows them to focus on competence instead of on matters of IT and infrastructure. Nevertheless, cloud computing has proven to have some limitations and disadvantages, especially for smaller business operations, particularly regarding security and downtime. Technical outages are inevitable and occur sometimes when cloud service providers (CSPs) become overwhelmed in the process of serving their clients. This may result to temporary business suspension. Since this technology's systems rely on the internet, an individual cannot be able to access their applications, server or data from the cloud during an outage.

# CHAPTER 2

## 2.1. Statement Formation of Problem

The ultimate goal of a statement of the problem is to transform a generalized problem (something that bothers you; a perceived lack) into a targeted, well-defined problem; one that can be resolved through focused research and careful decision-making.

Writing a statement of the problem should help you clearly identify the purpose of the research project you will propose. Often, the statement of the problem will also serve as the basis for the introductory section of your final proposal, directing your reader's attention quickly to the issues that your proposed project will address and providing the reader with a concise statement of the proposed project itself.

A statement of problem need not be long and elaborate: one page is more than enough for a good statement of problem.

Characteristics of a statement of the problem

A good research problem should have the following characteristics:

1. It should address a gap in knowledge.
2. It should be significant enough to contribute to the existing body of research
3. It should lead to further research
4. The problem should render itself to investigation through collection of data
5. It should be of interest to the researcher and suit his/her skills, time, and resources
6. The approach towards solving the problem should be ethical

## 2.2. Solution Approaches

Once the IT department has fully addressed these risk factors, they can move on to plan the best cloud migration approach to meet the company's business objectives and requirements. While there are a number of approaches used in the industry, below are the most broad:

**Lift and shift**: This approach involves mapping the on-premises hardware and/or VMs to similar resource-sized cloud instances. For example, if a company's front-end application server has 4 CPUs, 64GB of RAM, and 512GB of local storage, they would use a cloud instance that matches that configuration as closely as possible. The challenges with this approach is that on-premise solutions are typically over-provisioned with respect to resources in order to meet peak loads as they lack the elastic, auto-scaling features of cloud. This results in increased cloud costs, which may be fine if this is a short-term approach

**Refactor and rearchitect:** In order to best maximize the features of cloud, such as auto-scaling, migration can be the forcing function to take some time and re-architect the application to be more per formant and also keep the costs under control. It is also a good time to re-evaluate technology choices, as a company may be able to switch some solutions from more expensive commercial ones, to open-source or cloud-native offerings.

**Shelve and spend**: This third approach involves retiring a monolithic on-premises application and moving to a SaaS solution. An example of this would be an HCM (Human Capital Management) application, which is often times a disparate set of code bases tied together with a relational database, migrating to an offering such as Workday HCM. This allows the modernization of business logic and offloads the operational burden of the service and infrastructure to the SaaS provider.

While there are a number of hurdles and challenges to overcome when it comes to cloud migration, these approaches can ensure that CIOs and CSOs take the best route in order to capitalize on the benefits of moving to the cloud, while minimizing risk at the same time.

## 2.3. Introduction of Elliptic curve Cryptography

Elliptic Curve Cryptography (ECC) was first proposed by victor Miller and independently by Neal Koblitz in the mid- 1980s and has evolved into a mature public-key cryptosystem. Compared to its traditional counterparts, ECC offers the same level of security using much smaller keys. This result in faster computations and savings in memory, power and bandwidth those are especially important in constrained environments. More significantly, the advantage of ECC over its competitors increases, as the security needs increase over time. Recently the National Institute of standards and Technology (NIST) approved ECC for use by the U.S government. Several standards organizations, such as Institute of Electrical & Electronics Engineers (IEEE), American National Standards Institute (ANSI), Open Mobile Alliance (OMA) and Internet Engineering Task Force (IETF), have ongoing efforts to include ECC as a required or recommended security mechanism. Here we present our new algorithm using Diffie Hellman key exchange algorithm providing forward secrecy for web browsers application.

The history of cryptography is long and interesting. It has a very considerable turning point when two researchers from Stanford, Whitfield Diffie and Martin Hellman, published the paper ―New Directions in Cryptography‖ in 1976. They preface the new idea of public key cryptography in the paper. Public-key cryptography and symmetric-key cryptography are two main categories of cryptography. The Well-known public-key cryptography algorithms are RSA (Rivest, et al. 1978), El-Gamal and Elliptic Curve Cryptography. Presently, there are only three problems of public key cryptosystems that are considered to be both secure and effective (CARICOM, 2001). Table 1.1 shows these mathematical problems and the cryptosystems that rely on such problems.

| | **Mathematical problem** | **Detail** | **Cryptosystem** |
|---|---|---|---|
| 1 | Integer Factorization problem (IFP) | Given an integer n find its prime factorization | RSA |
| 2 | Discrete Logarithm problem (DLS) | Given integer g and h find x‘such that =gxmod n | Diffie-Hellman (DH) |
| 3 | Elliptic curve discrete logarithmic problem (ECDLP) | Given points P and Q on the curve find x ‘such that Q=xP | Diffie-Hellman (DH) |

**Table: 2.3**

### 2.3.1 Proposed Model

This section discusses briefly the use of Elliptic Curve Cryptography (ECC) a public key cryptography. In public key cryptography each user or the device taking part in the communication generally have a pair of keys, a public key and a private key, and a set of operations associated with the keys to do the cryptographic operations. Only the particular user knows the private key whereas the public key is distributed to all users taking part in the communication. Some public key algorithm may require a set of predefined constants to be known by all the devices taking part in the communication. „Domain parameters‟ in ECC is an example of such constants. The mathematical operations of ECC is defined over the elliptic curve each value of the „a‟ and „b‟ gives a different elliptic curve. All points (x, y) which satisfies the above equation plus a point at infinity lies on the elliptic curve. The public key is a point in the curve and the private key is a random number. The public key is obtained by multiplying the private key with the generator point G in the curve. The generator point G, the curve parameters „a‟ and „b‟, together with few more constants constitutes the domain parameter of ECC. The EC domain parameters are explained in section 9. One main advantage of ECC is its small key size. A 160-bit key in ECC is considered to be as secured as 1024-bit key in RSA.

### Discrete Logarithm Problem

The security of ECC depends on the difficulty of Elliptic Curve Discrete Logarithm Problem. Let P and Q be two points on an elliptic curve such that $kP = Q$, where k is a scalar. Given P and Q, it is computationally infeasible to obtain k, if k is sufficiently large. k is the discrete logarithm of Q to the base P. Hence the main operation involved in ECC is point multiplication. i.e. multiplication of a scalar k with any point P on the curve to obtain another point Q on the curve. **Pre-requisites to understand working of ECC:** Let P and Q be two points on the elliptic curve

**Pre-requisites to understand working of ECC:** Let P and Q be two points on the elliptic curve

- ADDING DISTINCT POINTS P AND Q
  $P = (xP,yP)$ and $Q = (xQ,yQ)$ are not negative of each other,
  $P + Q = R$ where $s = (yP - yQ) / (xP - xQ)$ $xR = s2 - xP - xQ$ and $yR = -yP + s(xP - xR)$
  Note that s is the slope of the line through P and Q.

- DOUBLING THE POINT P
  When yP is not 0, $2P = R$ where $s = (3xP2 + a) / (2yP )$ $xR = s2 - 2xP$ and $yR = -yP + s(xP - xR)$

### ECC Domain Parameters

Elliptic curve parameters over the finite field Fp or F2m can be described by one set tuple: $T = (q, FR, a, b, G, n, h)$ • q: the prime p or 2m that defines the field and at the same time decides the curve form; • FR: the field representation, i.e., using which method to represent the elements in the field (polynomial basis or normal basis or subfield basis for F2m, Montgomery residue for Fp); • a, b: the curve coefficient, depending on the security requirement; • G: the base point also known as the generator point, $G = (Gx, Gy)$, • n: prime order of G ie. n is the smallest prime number such that $nG=\infty$) • h: cofactor ie. Number of points over the curve-it should be as small as possible.

### Dealing with confidentiality aspect.

**Key Generation** Key generation is an important part where we have to generate both public key and private key. The client will be encrypting the message with his public key and upload on cloud. When required client can download and decrypt the message using his private key.
1. Select a random integer d such that $1\leq d\leq n-1$.
2. Compute $Q = dG$; d is the random private number, Q is the public key and G is the generator point on the curve. **Encryption**

Let „m‟ be the message that we are sending. We have to represent this message on the curve. This have in-depth implementation details. All the advance research on ECC is done by a company called Certicom.

**Encryption**

Let „m‟ be the message that we are sending. We have to represent this message on the curve. This have in-depth implementation details. All the advance research on ECC is done by a company called certicom.
Consider „m‟ has the point „M‟ on the curve „E‟. Randomly select „d‟ from [1 – (n-1)].
Two cipher texts will be generated let it be C1 and C2.

C1 and C2 will be send

C1=d*P

C2=M+d*Q

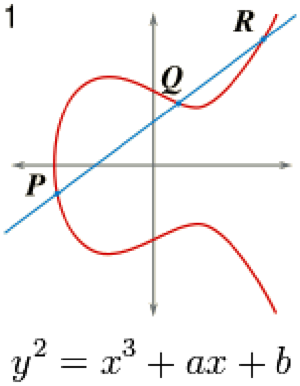


Fig 2.3.1: Standard ECC curve

**Decryption**

We have to get back the message „m‟ that was send to us,
M-C2-d*C1
M is the original message that we have send.

**B. Dealing with integrity aspect.**

**SHA-512**

SHA-512 is a set of cryptographic hash functions. Cryptographic hash functions consists of mathematical operations that run on digital data by comparing the computed hash value generated by the algorithm to a known or expected hash value. With the help of this one can determine the data"s integrity. In the proposed model, we are ensuring data integrity with the help of SHA-512 algorithm. This algorithm will be generating a hash value for the encrypted data which will be compared with the hash value of the corresponding data stored on cloud whenever the TPA wishes to perform data integrity check. If the generated hash value do not match with the data"s hash value that means the data is been modified and the same will be notified to the user. If the hash value matches that means that the data is secured and integrity is maintained.

## 2.3.2 System architecture

**System Model**

Cloud Data Storage Model The cloud storage model considering here is consists of three main components as illustrated in Fig.

1) **Cloud User**: The user, who can be an individual or an organization originally storing their data in cloud and accessing the data.

2**) Cloud Service Provider (CSP)**: The CSP, who manages cloud servers (CSs) and provides a paid storage space on its infrastructure to users as a service.

3) **Third Party Auditor (TPA) or Verifier**: the TPA or Verifier, who has expertise and capabilities that users may not have and verifies the integrity of outsourced data in cloud on behalf of users. Based on the audit result, the TPA could release an audit report to user.

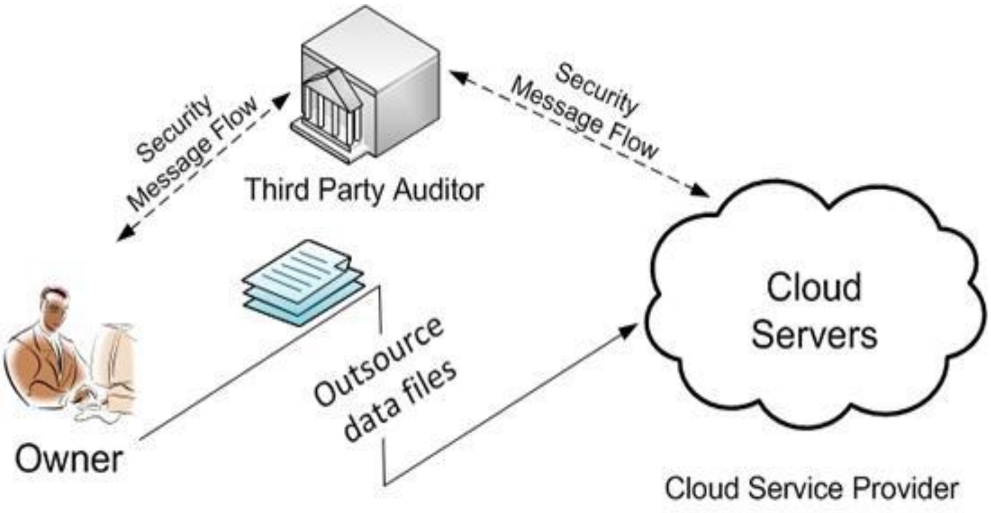
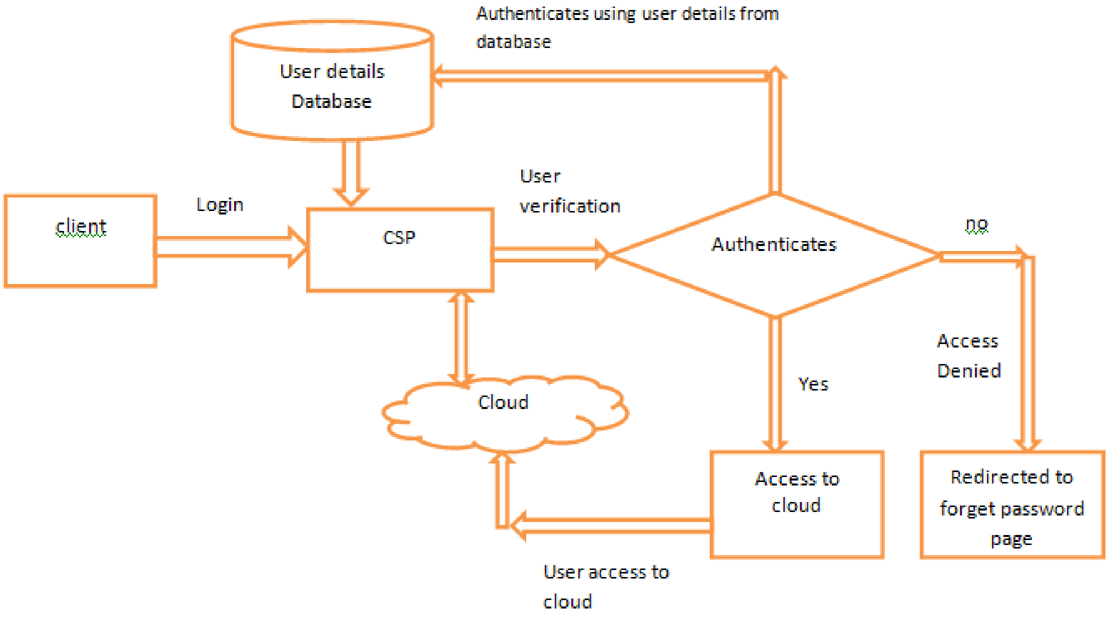
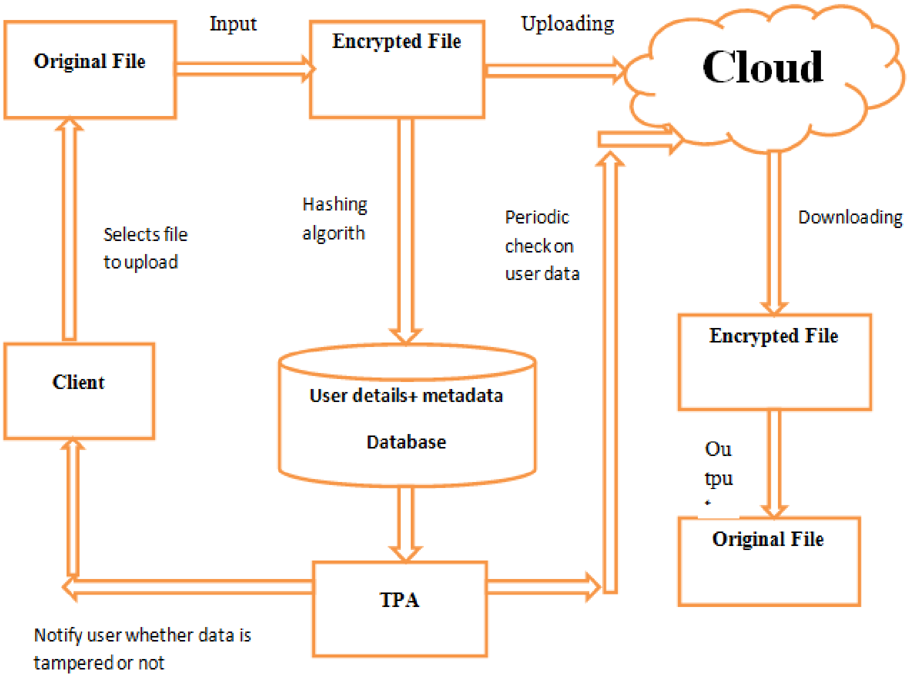


Fig2.3.2: System architecture

our system is designed to ensure security, confidentiality and integrity of data that is stored on cloud. First of all, the client who wants to store his data on cloud will register himself to the Cloud Service Provider (CSP) by providing all details. CSP will be then store all the details of the client like his id and password in the database of its own and auto-generate public and private key using ECC algorithm for each client which will be managed by an independent entity i.e. key repository. Generation of public key and private key is a backhand process Client will then login at cloud using his id and password after which initiation of file upload process will take place. Encryption of file chosen by client at the time of upload takes place using client's public key. This encrypted file will be stored on the cloud in the form of blocks. Whenever the user wishes to download the file, he will inform the CSP after which the file gets decrypted with the help of private key stored at key repository and the client gets back his original file. Another important component of the proposed system is Third Party Auditor(TPA). TPA stores the metadata of the file in its database which is obtained by applying MD5 hashing algorithm on encrypted blocks of file. TPA performs integrity check on the data periodically by comparing the checksum value of the blocks of data stored at cloud with metadata at TPA‟s database. If the value matches that means the data is secured and integrity of the data is maintained. However the TPA notifies the client each time irrespective of positive or negative result.



2.3.2.1: Schematic diagram showing login process



## Analysis

- Short key size: ECC employs a relatively short encryption key, a value that must be fed into the encryption algorithm to decode an encrypted message. This short key is faster and requires less computing power than other algorithms. For example, a 160-bit ECC encryption key provides the same security as a 1024-bit RSA encryption key and can be up to 15 times faster, depending on the platform on which it is implemented.

- More Complex: In spite of multiplication or exponentiation in finite field, ECC uses scalar multiplication. Solving Q= dP (utilized by ECC) is more difficult than solving factorization (used by RSA) and discrete logarithm (used by Diffie-Hellman (DH), EIGamal, Digital Signature Algorithm (DSA)).

- Power Consumption: ECC requires less power for its functioning so it is more suitable for low power applications such as handheld and mobile devices.

- Computational Efficiency: Implementing scalar multiplication in software and hardware is much more feasible than performing multiplications or exponentiations in them. As ECC makes use of scalar multiplications so it is much more computationally efficient than RSA and Diffie-Hellman (DH) public schemes. So we can say without any doubt that ECC is the stronger and the faster (efficient) amongst the present techniques.

## 2.4. Introduction of Diffie-Hellman

Diffie-Hellman key exchange is a special method of exchanging cryptographic keys. It is one of the earliest practical illustrations of key exchange implemented within the field of cryptography. The Diffie-Hellman key exchange system allows 2 parties that have no prior knowledge of each other to jointly establish a contribution secret key over an insecure communications channel. This key can then be used to encrypt subsequent mediums using a symmetric key cipher.

The scheme was 1st published by Whitfield Diffie and Martin Hellman in 1976, although it had been individual invented a few years earlier within GCHQ, the British signals intelligence organization, by Malcolm J. Williamson but was kept classified. In 2002, Hellman suggested the algorithm be called Diffie-Hellman-Merkle key interchange in recognizance of Ralph Merkle's deposition to the invention of public key cryptography. Although Diffie-Hellman key agreement itself is a non-authenticated key-agreement protocol, it provides the backbone for a variety of authenticated protocols, and is used to provide perfect forward Elliptic curve Cryptography and Diffie-Hellman Key exchange confidentiality in Transport Layer Security's temporary modes.

### 2.4.1 DESCRIPTION

Diffie–Hellman establishes a shared secret that can be used for secret mediums by exchanging data over a public network. The following structure illustrates the common idea of the key exchange by using colors instead of a larger number. The key part of the process is that A and B exchange

Their secret colors in a mixture only. Finally, this generates a unique key that is mathematically difficult (impossible for advance supercomputers to do in a reasonable amount of time) to reverse for another party that might have been listening in on them. X and Y now use this common secret to encrypt and decrypt their sent and received data. Note that the yellow paint is already agreed by X and Y.

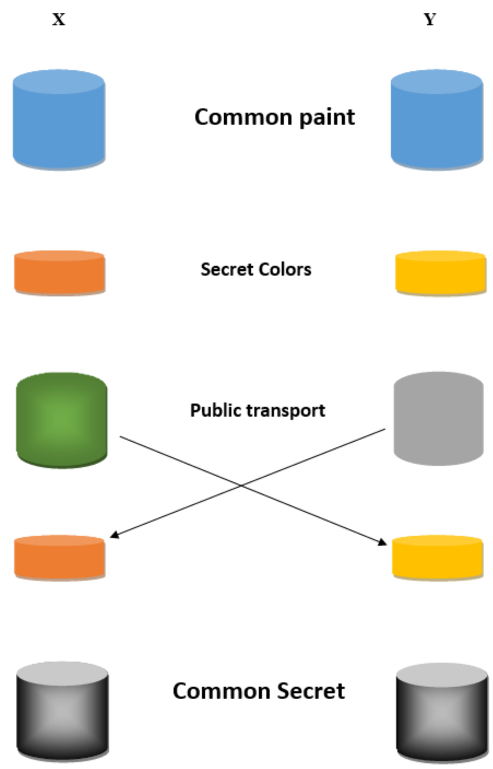
### 2.4.2 Security against Attacks

The basic Diffie-Hellman protocol we have shown is not secure against a Man-in-the-middle attack. In fact, impossible to achieve security against such an attacker unless some information is shared in advance E.g., private key setting Or public key setting. To address this issue, generally a process of authentication will be expected to guarantee that, at whatever point Alice wishes to send a message to Bob, the beneciary must be Bob and not an Eve, and the other way around. It is also important - and generally the norm - to discard the keys after use, so that there will be no long - term keys that can be revealed to bring about issues later on. Deferent concerns ordinarily rotate around upgrading the mathematics involved. That is, to properly generate the randomly choose values with the goal that they are large enough to achieve computational infeasibility for attackers, and random enough, as pseudo- random numbers can greatly ease Eve due to their eventual predictability. "Generally talking, the fundamental thought is as per the following. Preceding execution of the protocol, the two gatherings Alice and Bob each acquire an public/private key pair and a certificate for the public key. During the protocol, Alice computes a signature on certain messages, covering the public value ga mod p

Bob proceeds in a similar way. Despite the fact that Eve is still ready to intercept messages in the middle of Alice and Bob, she can't forge signatures without Alice's private key and Bob's private key. Henceforth, the upgraded enhanced protocol defeats the man-in-the-middle attack."

### 2.4.3 Advantages and Disadvantages

The security factors with respect to the fact that solving the discrete logarithm is very challenging, and that the shared key (i.e. the secret) is never itself transmitted over the channel. The algorithm has its share of drawbacks including. The fact that there are expensive exponential operations involved, and the algorithm cannot be used to encrypt messages - it can be used for establishing a secret key only.



DIFFIE-HELLMAN KEY EXCHANGE

## 2.5. Algorithm

X and Y Compute edB = S = (S1, S2). (Using Diffie – Hellman Scheme)

X sends a message M ε E to Y as follows: Compute (S1 * S2) mod N = K.

Compute K*M = C, and send C to Y.

Y receives C and decrypts it as follows: Compute (S1 * S2) modN = K.

Compute (K–1) modN.

(where N = #E)

K–1*C = K–1*K*M = M.

In the first method (M1), the sender computes the multiplication between the coordinates of the key in the encryption algorithm, and the receiver compute the multiplication between the coordinates of the key in the decryption algorithm.

Algorithm of (M2)

X and Y Compute edB = S = (S1, S2).

Using Diffie – Hellman Scheme)

X sends a message M to Y as follows: Compute (s1

S2) mod N = K.

Compute K*M = C, and send C to Y.

Y receives C and decrypts it as follows: Compute (s1

S2) mod N = K.

Compute (K–1) mod N.

K–1*C = K–1*K*M = M.

In the second method (M2), we support the system more security of the first method, because the sendercompute the exponentiation function between the coordinates of the key in the encryption algorithm (use fast exponentiation method), and the receiver compute the inverse of the exponentiation function between the coordinates of the key in the decryption algorithm.

## CHAPTER 3

### 3.1 Implementation

Let E be an elliptic curve define over Where p = 3023 with parameters a = 1, b = 2547Where (4a3+27 b2) mod p = 2027 0.
And #E = 3083.
Since #E is prime number then by theorem1, every point on Ein base point, therefore let B = (2237, 2480).
To apply this system test using (M1), at first we must apply Diffie-Hellman Exchanging key
X chooses a secret random integer e = 2313.
eB = 2313 (2237 , 2480) = (934 , 29)And send (934, 29) to Y .
Y chooses a secret random integer d = 1236.
dB =1236 (2237 , 2480) = (1713, 1709)And send (1713 , 1709) to X
X computes the secret key e (dB)= 2313 (1713, 1709).
edB = (2537 , 1632) = S
Y computes the secret key d (eB) =1236 (934,29).
deB = (2537 , 1632) = S
Now, X and Y have the same point S = (2537, 1632)If X send a message M = (2284, 2430) to Y Compute (S1, S2).mod p = (2537 * 1632) mod 3083= 2998 = K.
Compute K*M = 2998 (2284, 2430)
= (2179, 1833)
=C, and send it to Y.
Y receives C and decrypts it as follows= Compute (S1, S2).mod p = 2998 =K
o Compute (K–1) mod N = (2998)–1mod 3083= 1342
o K–1 C =1342 (2179, 1833)
= (2284, 2430)
To apply this system test using the algorithm (M2), at first we must apply Diffie–Hellman Exchanging key.
By the same procedure to solve Diffie–Hellman scheme we have obtained
S = (2537, 1632)
If X sends a message M = (2284, 2430) to Y using(M2), he does the following:
Compute(s1
S2)mod N = (25371632) mod
3083= 323=K.
o Compute K* M = 323 (2284, 2430)= (2555, 1066)
=C, and send it to Y.
Y receives C and decrypts it as follows Compute (s1
S2)mod N = 323=K.
o Compute (K–1) mod N = (323)–1 mod3083= 1594.
o K–1 C = 1594 (2555 , 1066)= (2284, 2430) = M.

## 3.2 Results & Findings

Performance of Elliptical Cryptography with Diffie Hellman Key Exchange will depend on the hardware of system. The Performance of Elliptical Cryptography with Diffie Hellman Key Exchange will also depend on another factor that is the quality of the JavaScript which is our execution environment.
The following table shows the times taken for various public-key operations on a cross-section of browsers and hardware both.
EC multiply = Elliptic curve point multiplication, bit size denotes both curve prime size and scalar multiplier size.

#### Analysis of Test Results

#### Key Generation Time

In both systems the key generation times were not the same every time, even though the key length is the same and it can sometime a take very long time to generate the keys. Figure 2 shows that for smaller key sizes the key generation time is almost equal in both cases, but as the key size grows RSA takes more amount of time to generate the keys and this time increases exponentially by the key size.Fig2 shows the comparison of the key generation times for RSA and ECC.[1]

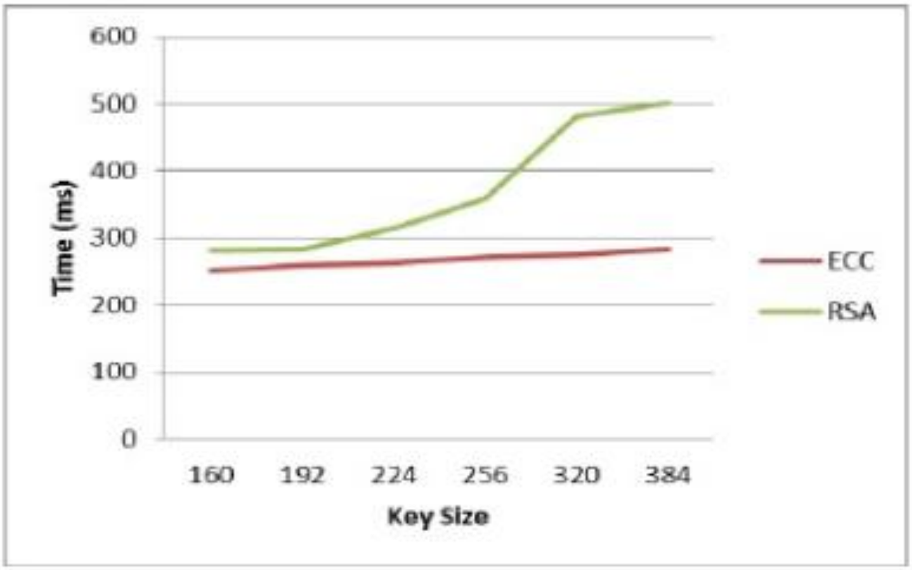


Figure 3.2.Comparison of Key Generation Time

### Encryption/Decryption Time

Figure 3 shows the Encryption times for ECC and RSA algorithms. Since JAVA implementation of RSA doesn’t support key sizes lesser than 512 bits length, simulation had to compare the encryption/decryption times between these two algorithms with different key sizes. Looking at the results, for smaller key sizes ECC provides much faster encryption/decryption as compared to RSA. Since RSA uses higher key sizes the encryption/decryption times grow exponentially with the given key size.[1]

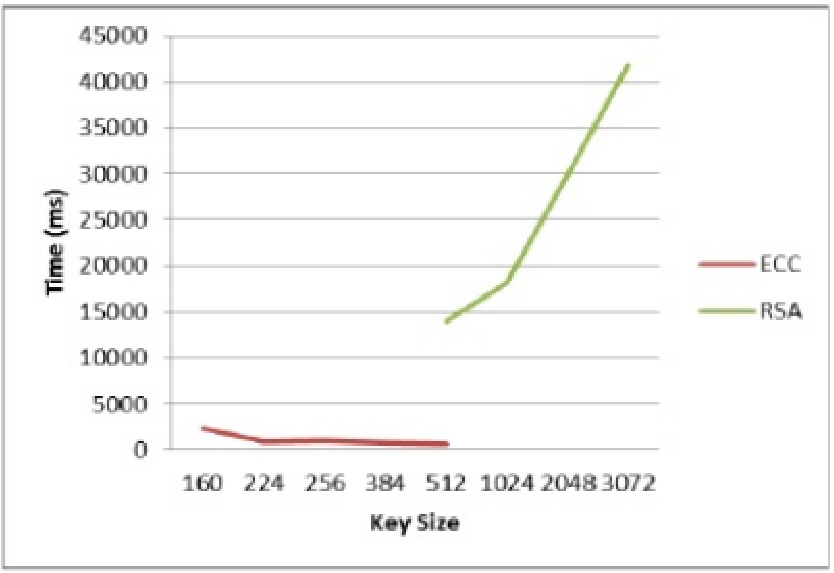


Figure 3.2.1Comparison of Encryption times

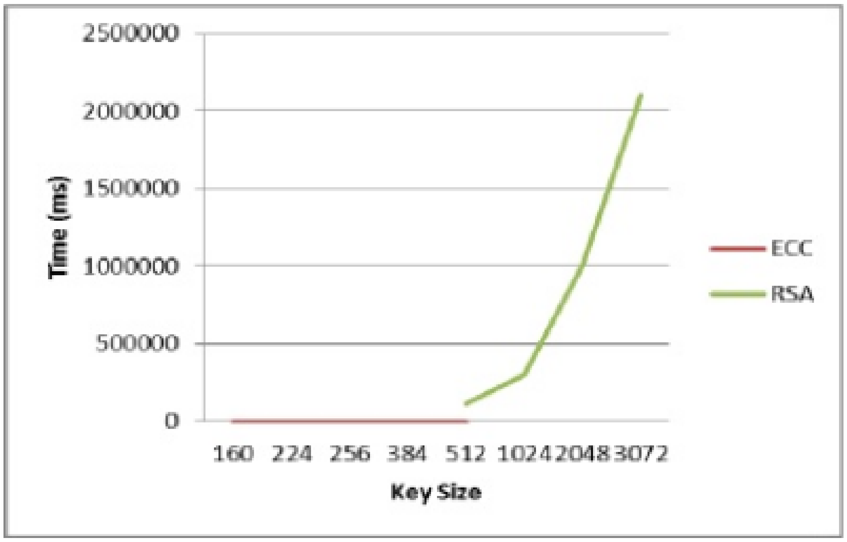


Figure 3.2.2.Comparison of Decryption times

### 3.3 Conclusion & Future Scope

Successful implementation of the projected system that is “Cloud Security framework supported code and compression technique and Diffie-Hellman protocol”. The projected technique concludes that the error rate is minimum then the previous paper. This technique consists of the protection in cloud that is handle by 2 keys that is schedule by ECC-DH that provides the entropy worth that is bigger than the opposite, which means this technique is safer than the opposite system. As per study, the previous papers didn’t specialize in Storage of cloud, which might be reduced via Huffman compression technique. this technique is said to the protection problems in cloud has been victorious implement with average error rate with 2 forms of strategies for storage and resource allocation in cloud computing. Within the projected system programming is finished with varied key size factors.

The typical Error rate is two.94% and therefore the Average worth of Entropy worth that's shaping the protection of encrypted information is ninety-three. In spite of the fact that Dffie-Hellman is an public key algorithm, specialists say it don't scale well for future. As of right now it is expressed that Diffie Hellman keys shorter than 900 bits are not secure. To make Deffie- Hellman keys, which now can go to 1,024 bits, secure for the following 10 to 20 years, associations would need to grow to key lengths of no less than 2,048 bits, as per Stephen Kent, chief researcher at BBN Technologies. In the long run, key sizes would need to grow to 4,096 bits. Researchers from the NIST's security technology group expect, that it is exceptionally conceivable, that Diffie-Hellman will be broken inside of 10 years or somewhere in the vicinity. The cryptographic security standards utilized as a part of public-key infrastructures, RSA and Dffie-Hellman, were presented in the 1970s. And although they haven't been broken, their time could be running out.

## 3.4 References

[1] Ravi Gharshi, Suresha “Enhancing Security in Cloud Storage using ECC Algorithm” International Journal of Science and Research (IJSR), India.

[2] PuneethaC1, Dr. M Dakshayini2 “Data Security in Cloud Using Elliptic Curve Cryptography” International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 5, May 2014.

[3] Dr.Chander Kant, Yogesh Sharma “Enhanced Security Architecture for Cloud Data Security” International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5, May 2013.

[4] Confidentiality, integrity, and availability (CIA triad):

http://whatis.techtarget.com/definition/Confidential ity-integrity-and-availability-CIA

[5] Wang. "Enterprise cloud service architectures".

[6] Jump up to:a b "What is Cloud Computing?". Amazon Web Services. 2013-03-19. Retrieved 2013-03-20.

[7] Baburajan, Rajani (2011-08-24). "The Rising Cloud Storage Market Opportunity Strengthens Vendors". It.tmcnet.com. Retrieved 2011-12-02.

[8]Oestreich, Ken (2010-11-15). "Converged Infrastructure". CTO Forum. Thectoforum.com. Archived from the original on 2012-01-13. Retrieved 2011-12-02.

[9] "Where's The Rub: Cloud Computing Hidden Costs". 2014-02-27. Retrieved 2014-07-14.

[10] "Cloud Computing: Clash of the clouds". The Economist. 2009-10-15. Retrieved 2009-11-03.

[11] "Gartner Says Cloud Computing Will Be As Influential As E-business". Gartner. Retrieved 2010-08-22.

[12]Gruman, Galen (2008-04-07). "What cloud computing really means". InfoWorld. Retrieved 2009-06-02.

[13] Jump up to:a b "Announcing Amazon Elastic Compute Cloud (Amazon EC2) - beta". 24 August 2006. Retrieved 31 May 2014.

[14] Antonio Regalado (31 October 2011). "Who Coined 'Cloud Computing'?". Technology Review. MIT. Retrieved 31 July 2013.

"Internet History 1977".

[15] "National Science Foundation, "Diagram of CSNET," 1981".

[16] Jayachander Surbiryala, Chunlei Li, ChunmingRong Department of Electrical Engineering and Computer Science University of Stavanger, Norway

http://www.in.idc.asia/ 2013 "LuitInfotech: What is Cloud Computing", http://www.luitinfotech.com/kc/what-is-cloud-computing.pdf

[17] Vani Dayal Sharma, Somya Aggarwal, Syeda Shira Moin, Mohammed Abdul Qadeer Department Of Computer Engineering, Zakir Hussain College of Engineering and Technology, Aligarh Muslim University, Aligarh 202002, India

http://searchservervirtualization.techtarget.com/definition/virtualization (retrieved in Feb 2013)

http://searchcloudcomputing.techtarget.com/definition/cloud-computing (accessed in Feb 2013)

[18] Martin Roesch,"SNORT — Insubstantial Intrusion Detection for Networks", In Proceedings of LISA '99: 13th Systems Administration Conference, pp. 229-238, November 7–12, 1999

[19] Stephen M. Specht, Ruby B. Lee," Dispersed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures". In Proceedings of the 17th International Conference on Parallel and Dispersed Computing Systems,2004 International Workshop on Security in Parallel and Distributed Systems, pp. 543-550, September 2004.

[20] Radwane Saad, FaridNait-Abdesselam and Ahmed Serhrouchni, "A Co-operative Peer-to-Peer Architecture to Defend Agiinst DDoS Attacks". In 33rd IEEE conference on local computer network, pp. 427-434, IEEE, September 2008

[21] Sebastian Roshke, Feng Cheng, Christoph Me`inel, Intrusion Detection in the Cloud". In Eighth IEEE Universal Conference on Dependable, Autonomic and Secure Computing, pp. - 729-734, IEEE, October 2009

[22] Aman Bokashi, Yogesh B, "Fortifying cloud from DDoS Attacks using Intrusion Detection System in Virtual Machine". In Second International Conference on Communication Software and Networks, IEEE, April 2010

# List of Publications

**1.** CLOUD SECURITY USING ELLIPTIC CURVE CRYPTOGRAPHY AND DIFFIE HELLMAN

# Curriculum Vitae

Full Name: Rishi Kumar

Father's Name: Ahivaran Lal

Date of Birth: 15th July 1995

Sex: Male

Nationality: Indian

Address: 356-SB/070 Sheela Garden, Alamnagar, Lucknow, U.P., India.

Phone Number: +919695010214

Email Address: rishi95715@gmail.com

## Academic Background

| Examination/Degree | Institution | Year of Passing | Percentage/CGPA |
|---|---|---|---|
| M.Tech. (CSE) | BBDU Lucknow | 2019(Appearing) | |
| B.Tech (CSE) | BBDEC Lucknow | 2017 | 64% |
| Class XII | U. P. Board | 2013 | 64% |
| Class X | U. P. Board | 2008 | 72% |

## Fields of interest

- Wireless Network and Network Security.

## Strengths

- Positive Attitude, Social Interaction, Hardworking.

## Interest and hobbies

- Watching Movie.
- Travelling.