DESIGN AND ANALYSIS OF FILTER BASED APPROACH TO IMPROVE THE DATA INTEGRITY IN SERVERLESS COMPUTING

A Thesis submitted

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY
IN
COMPUTER SCIENCE AND ENGINEERING

By
A.ARULPRAKASH
17SCSE301028

Supervisor

Dr. SAMPATH KUMAR K
Professor

SCHOOL OF COMPUTING SCIENCE & ENGINEERING
GALGOTIAS UNIVERSITY
Plot No 2, Sector 17-A Yamuna Expressway
Greater Noida, Uttar Pradesh
INDIA

2021

CANDIDATE DECLARATION

I hereby certify that the work which is being presented in the thesis, entitled "Design and analysis of filter based approach to improve the data integrity in serverless computing" in fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Faculty of Computer Science and Engineering and submitted in Galgotias University, Uttar Pradesh is an authentic record of my own work carried out during a period from January 2018 under the supervision of Dr. Sampath Kumar K, Professor, SCSE.

The matter embodied in this thesis has not been submitted by me for the award of any other degree or any other University/Institute.

(A. ARULPRAKASH)

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

(Dr. Sampath Kumar K)

Supervisor

School of Computing Science and Engineering

The Ph.D. Viva-Voice examination of **A.Arulprakash** Research Scholar, has been held on_____.

Sign. of Supervisor Sign. of External Examiner

Galgotias University
Uttar Pradesh
School of Computing Science & Engineering

CERTIFICATE

This is to certify that **Mr. A. ARULPRAKASH** has presented his pre-submission seminar of the thesis entitled **"DESIGN AND ANALYSIS OF FILTER BASED APPROACH TO IMPROVE THE DATA INTEGRITY IN SERVERLESS COMPUTING"** before the committee and summary is approved and forwarded to School Research Committee of School of Computing Science & Engineering, in the Faculty of Engineering & Technology, Galgotias University, Uttar Pradesh.

Dr. SAMPATH KUMAR K
Supervisor & Professor,
School of Computing Science & Engineering
Galgotias University
Uttar Pradesh

Dean
School of Computing Science & Engineering
Galgotias University
Uttar Pradesh

Dean PG & PhD
Galgotias University
Uttar Pradesh

Date :

STATEMENT OF THESIS PREPARATION

1. Thesis title: Design and Analysis of Filter Based Approach to Improve the Data Integrity in Serverless Computing
2. Degree for which the thesis is submitted: Doctor of Philosophy in CSE
3. Thesis Guide was referred for preparing the thesis.
4. Specifications regarding thesis format have been closely followed.
5. The contents of the thesis have been organized based on the guidelines.
6. The thesis has been prepared without resorting to plagiarism.
7. All sources used have been cited appropriately.
8. The thesis has not been submitted elsewhere for a degree.

(Signature of the student)

Name: A. ARULPRAKASH

Roll No. 17SCSE301028

APPROVAL SHEET

This thesis/dissertation/report entitled Design and Analysis of Filter Based Approach to Improve the Data Integrity in Serverless Computing by A. ARULPRAKASH is approved for the degree of Doctor of Philosophy

Examiners

_____

_____

Supervisor (s)

_____

_____

_____

Chairman

_____

Date:

Place:

ABSTRACT

Serverless computing has supported new and persuasive frameworks for serverless applications in the current trends to move the container and microservice applications. This is an indication of a growing emphasis on corporate events, conferences, blogging and development on serverless computing. In the scholarly community, however, the enthusiasm was minimal.

In the view of an IaaS client, this paradigm shift is both a possibility and a threat. First of all, it provides developers a simplified programming model that builds serverless applications to remove any of all operational issues, reduce the costs for serverless code by charging for run-time rather than the allocation of resources and quickly deploy a small portion of the un-server code, for example, to respond to accidents.

Serverless computing provides new advantages from a serverless platform to handle an entire development stack, decreasing maintenance costs by optimized serverless resource management; Providing a platform for more network services and reducing the operating power required to create and sustain serverless applications.

Serverless computing is a principle of the industry that describes the framework and architecture of programming where small code fragments are done and then the resource code executes are managed. There is no suggestion that the developer really can encourage the serverless provider to deal with much of the operational problems, including services, logging, support, scalability and tolerance for defects.

Serverless systems promise's a new feature which allows a scalable microservices to be written and save costs. The next step is the development of serverless computing architectures. The successful implementation of serverless can be challenging because the term overlaps with other definitions like Platform as a Service (PaaS) and software as a service (SaaS). The Infrastructure as a Service (IaaS) is where the developer most efficiently handles the application code and operating infrastructure in the serverless world. Here the developer is liable and can customize some aspects of the implementing a software. It is supported with applications or virtual machines.

In the first part of the research study, an identity based cryptosystem for secured authentication is designed that uses Eliptic Curve Cryptography (ECC) model for securing the data in serverless environment. The segregation of initialization phase and authentication phase that enables the ECC model to improves the security and authentication of the data in serverless environment.

An attribute based encryption is designed in the second part of the research study under different security requirements that includes data confidentiality, collision resistance, attack resistance, non-accessibility of sensitive before release time and delete data after expiration time. Considering all these parameters, the encryption model offers improved security of data that gets traversed or stored in the serverless environment.

As a final part of the research study, a filter based security approach is proposed that undergoes three different mechanisms including encrypted fuzzy based filter with link reliability estimation, VM reliability estimation and residual energy factor estimation. Secondly, it includes the adoption of signature verification scheme involving encryption and decryption mechanism.

Thus an experimental analysis are conducted on all the three proposed models identity based cryptosystem, attribute based encryption and filter based security. The experimental analysis is conducted in terms of different security metrics that includes encoded key size, signature time and key generation time in terms of its minimum, average and maximum time. Further, it is tested in terms of serverless network metrics that includes delay time, average throughput, average response time, error rate, load distribution and cost-efficiency.

The results of simulation on serverless network metrics shows that the filter based security using fuzzy logic achieves higher average throughput and reduced delay, average response, error rate and load distribution when compared with attribute based encryption and identity based cryptosystem. Further, the testing on security metrics shows that the filter based security using fuzzy obtains reduced encoded key size, signature time and key generation time with reduced minimum, average and maximum time than attribute based encryption and identity based cryptosystem. Further, the study shows that the filter based security using fuzzy is cost-efficient.

ACKNOWLEDGEMENT

Working as an Assistant Professor and doing research for the degree of Ph.D in Galgotias University was quite magnificent and challenging experience for me. In all these years, many people directly or indirectly contributed in shaping up my career. It was hardly possible for me to complete my doctoral work without the precious and invaluable support of these personalities.

I would like to give my small tribute to all those people. Initially, I would express my sincere gratitude to my supervisor Dr.SAMPATH KUMAR K Professor, School of Computing Science and Engineering for his valuable guidance, enthusiasm and overfriendly nature that helped me a lot to complete my research work in a timely manner.

I must owe a special debt of gratitude to Hon'ble Chancellor Mr. Suneel Galgtotia, Mr. Dhruv Galgotia, CEO and Hon'ble Vice-Chancellor Dr. Preeti Bajaj, Galgotias University for their valuable support throughout my research work.

I express my sincere thanks to Dr. Munish Sabharwal, Dean School of Computing Science & Engineering and Dr.Naresh Kumar, Dean PG & PhD for their guidance and moral support during my research work and all faculties of School of Computing Science & Engineering who helped me a lot in my course of research work and all those who stood behind me.

Nothing is possible without the constant support of my family. I would like to convey my deep regard to my parents for their wise counsel and indispensable advice that always encouraged me to work hard for the completion of my research work. My highest gratitude goes to my parent's and all my family members for their relentless support, blessings and encouragement. Special mention goes to my wife, A. Ishwarya and my kids, My final thanks to all my friends, and to all those who stood behind by me like a support and helped me in completing this dissertation.

Mr.A.ARULPRAKASH

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SOA	Service-Oriented Architecture
P2P	Peer-to-peer
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
FaaS	Function as a Service
BaaS	Backend as a Service
AWS	Amazon Web Services
CGI	Common Gateway Interface
API	Application Program Interface
PKI	Public Key Infrastructure
MAC's	Message Authentication Codes
HMAC's	Hashed-MAC's
VM	Virtual Machine
CDC	content-defined chunking algorithm
ABR	Attribute Based Revocation
CP-ABE	Ciphertext-Policy Attribute-Based Encryption
DURKR	Dynamic User Revocation & Key Refreshing
ECC	Elliptic Curve Cryptography
SSP	Serverless Service Provider
HMAC	Hash-based Message Authentication Code
ADS	ABE-based secure document self-destruction scheme
DSC	Data Storage Center
KGC	Key Generation Center
SCSD	A Secure Ciphertext Self-Destruction scheme
EDO	Encapsulated self-Destruction Object
EPCDD	Efficient and Privacy- preserving Big Data Deduplication strategy

TPA	Third Party Auditor
PoW	Proof of Ownership
CE	Convergent Encryption
PP-CLPDP	Privacy-Preserving certificateless Provable Data Possession
PDP	Provable data possession
MR-PDP	Multiple –replica PDP
HLA	homomorphic linear authenticator
PRP	Pseudorandom Permutation
PRFs	pseudo random function
HAIL	A high availability and integrity layer
OISAP	Optimal Intra need cluster share allocation problem
SSS	Short secret scheme
IDA	Information dispersal algorithms
MHT	Merkle Hash tree
PVP	public verification parameter
IHT	Index –hash table
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDLP	Elliptic curve discrete logarithm problem"
PBC	Pairing-Based computational
PSEC	Probably Secure Encryption Curve
CRC	Cyclic Redundancy Check
CBR	Constant bit rate
BDHE	Bilinear Diffie-Hellman Exponent
ICT	Information Communications Technology
MS	Milli Second
KBPS	Kilobytes per second
KDF	Key derivation Hash function
AES	Symmetric-key Encryption

LIST OF PUBLICATIONS

PUBLISHED IN JOURNAL:

- A.Arulprakash,V.Arul, S.Jagannathan, "A Look at of Efficient and more Suitable Load Balancing Algorithms in Cloud Computing" in International Journal of Engineering Research in Computer Science and Engineering, Volume 5, Issue 4, April 2018.
- A.Arulprakash, R.Viswanathan, "Preservation Of Iris Biometrics In Cloud With Cloud Id Screen Algorithm" in International Journal of Scientific & Technology Research, volume 9, issue 02, February 2020 (Scopus).
- A.Arulprakash, Dr.K.SampathKumar, "Improved Encryption Towards Data Security in Serverless Computing" in Journal of Computational and Theoretical Nanoscience, Volume 17, Number 12 , May 2020 (Scopus) .
- A.Arulprakash, Dr.K.SampathKumar, "Design And Analysis Of Server less Security Using Identity Based Cryptosystem for Secured Authentication" in Journal of Advanced Research in Dynamical and Control Systems ,Volume 12 | 06-Special Issue, August 2020 (Scopus) .
- A.Arulprakash, Dr.K.SampathKumar, "Improved Data Integrity in Cloud Serverless Environment Using Filter Based Approach" in Solid State Technology, Volume 64, No. 2 (2021) (Scopus) .

PRESENTED IN CONFERENCE:

- A.Arulprakash, R.Viswanathan, "Information Security in Cloud Computing: A Survey" in National conference on Recent Challenges in Engineering Science and Technology NCRCEST-2019.
- A.Arulprakash, Dr.K.SampathKumar, "Cloud Serverless Security and services : A Survey" in 2nd Innovative Product Design and intelligent Manufacturing systems: National Conference (IPDIMS 2020) (SPRINGER)

CHAPTER -I

INTRODUCTION

CHAPTER - I

INTRODUCTION

1.1. CLOUD COMPUTING

Cloud computing controls the entire information technology (IT) by allowing omnipresent access to a distributed pool of configurable device tools and a higher degree of limited managed services. The allocation of resources and the cost-effectiveness of these resources is a prerequisite for this process. In today's scenario, building a cloud architecture is easily reliant based on the availability of high capability networks, low-cost computers, storage devices like hardware virtualization with a service-driven architecture and self-employed utilities [1].

The word of cloud is simply a forum for distributed Internet-based computer resources. The aim of cloud computing is to help consumers to achieve these advantages through pay-as-you-go access to all technologies. The customer of the cloud services does not even need to have a deep knowledge of the services. The cloud is based on the Service-Oriented Architecture (SOA) principle which helps the consumer to avoid business problems through the integration of these cloud services. This permits the proper use of its facilities and services under a well-defined platform, allowing for a standardized use of good practice in the SOA sector. The cloud offers data generation and intensive parallel applications with software and innovations at a far more manageable cost, than any standard parallel computing technology [2]. Figure 1.1 illustrates the general cloud computing model.

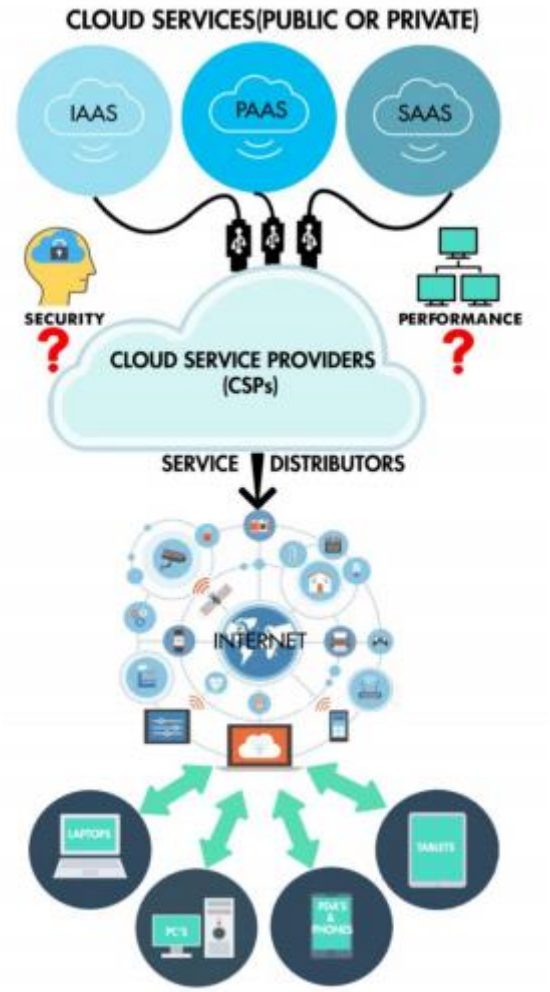


Figure 1.1 Metaphor on Cloud Computing

Different cloud computer services and templates are given as follows:

- Client-server model: Client-Server Computing refers to request and response data or information that occurs in a distributed network with different functionality between each service provider on various nodes.
- Computer bureau: A facility that gives quick and instantaneous access within a small or private network to computer services.
- Grid computing: Grid is a vast set of resources which function towards a shared purpose. It has advanced clustered and parallel computing functions, in which super and virtual computers consist of a cluster of networked and loose-coupled computers operating on a larger scale.

- Fog computing: Fog often represents a distributed computing model that provides customers or edge consumers closer to the network with data calculation, storage and device resources. Instead of sending the data to some remote location for processing, it also manages data at network level, the end-user client, and any smart devices.
- Mainframe computer: These are the machines with huge volumes of data analysis that support businesses and large corporate environments in sensitive data applications.
- Utility computing: This is a full package of on-demand computing services open to consumers. Utility computing is a huge backbone for cloud computing growth.
- Peer-to-peer: P2P itself is a distributed architecture with no centralized data sharing coordination. A producer and user of the resource is a participant.
- Green computing: It eliminates environmental impacts on IT activities with renewable and energy-efficient machines to ensure economic viability.
- Cloud sandbox: Highly efficient technology with a shared infrastructure creation, automation and movement design.

The various services offerings of Cloud providers providers are:

- Google: It provides a private cloud to provide Google documentation, as well as many other users' features, such as email access, implementations with documentation, text translations, maps, web analytics, etc.
- Microsoft: Microsoft Office 365 is an online service that provides a cloud computing tool for content and business analytics.
- Salesforce.com: A Software-as-a-Service model that offers its customers strong client relationships. Force.com and Vmforce.com provide developers with a forum to build personalized cloud applications.

Characteristics of Cloud Computing

Cloud computing has a number of features that are as follows:

- Distributed infrastructure: It offers a virtualized architecture of computing for distributing physical resources, storage and networking on the Internet.
- Dynamic provisioning: It allows commodities to be delivered in line with the existing demand, which is automatically accomplished by the implementation of software automation. This allows dynamic scaling and building of highly reliable and secure infrastructure capabilities.
- Network access: This allows access to the Internet via a standard application programming interface to a broad variety of devices such as PCs, laptops and mobile networks (API).
- Managed metering: This enables metered monitoring and usage of facilities in order to maximize use of utilities and provide information for accurate monitoring and billing.

Service Models in Cloud

A cloud environment is created, the computer resources are distributed in different business models. These models can be dependent on the needs of user market plans and various functionalities. These models are shown on different levels in Figure 1.2.

Figure 1.2 Cloud Computing Service Models

In every cloud field, the three standard models are as follows:

- Infrastructure as a Service (IaaS): Consumers have the ability to configure and monitor the operating system level, device level, storage level and system network access. But they can't monitor the cloud architecture themselves. It offers simple access to basic services, such as physical computers, virtual machines, virtual storage, etc.

- Platform as a Service (PaaS): It enables consumers to buy and distribute their own cloud apps and services across the networks. You may not have to think about operating system maintenance and network connectivity, but the service has certain limitations on program rollout. It provides the whole runtime environment and the programming resources required for development of applications.

- Software as a Service (SaaS): Users may acquire services in order to connect and to use an application or other cloud services. It enables technology programs to be used as a full service kit by end-users.

Deployment Models in Cloud

Cloud deployment is dependent entirely on users' specifications. The following are four basic implementation models and features supporting different customer needs in the cloud. The pictorial depiction of these forms is clearly seen in Figure 1.3.

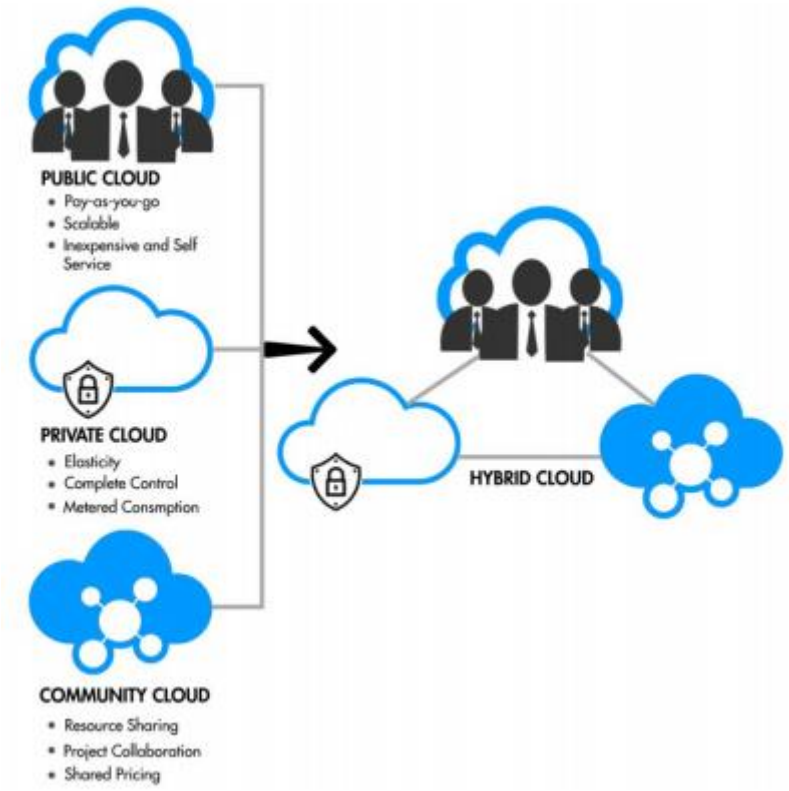


Figure 1.3 Types of Deployment Models in Cloud Computing

Private Cloud:

Private clouds are designed by just one business. The objective of these forms is to tackle data protection problems and to have more autonomy, which is usually a public cloud defect.

Public Cloud:

Third-party providers own and serve public clouds. They provide consumers with excellent, attractively affordable and cost-effective infrastructure facilities. Both users are distributing an architecture pool close to the cloud service provider that contains a small setup, security safeguards, and compatibility variations. This infrastructure is typically bigger than the cloud of businesses, so it is able to grow on demand effortlessly.

Hybrid Cloud:

The hybrid clouds, which combine both public and private cloud models, With hybrid cloud, service providers may either completely or partially use third-party cloud providers so that storage is more flexible. The hybrid cloud environment is capable of providing its customers with an external on-demand scale. This mix increases public cloud privacy resources, making it much easier to handle any sudden workload spikes than most cloud forms.

Community Cloud:

Here a variety of organisations with common goals and needs are spread over the cloud networks. This is extremely useful because it reduces the cost of capital investment for the company to be split by the organisations. The procedure may be carried out in-house or on the premises with a third party.

Architectural Representation of Cloud

Cloud architecture refers to the outline of a device or the skeletal view of a virtual system that provides cloud storage services. This usually includes various cloud components that communicate loosely via a loosely linked mechanism. Elastic provisioning based on relevant scenarios determines intelligence in the use of close or losing link structures.

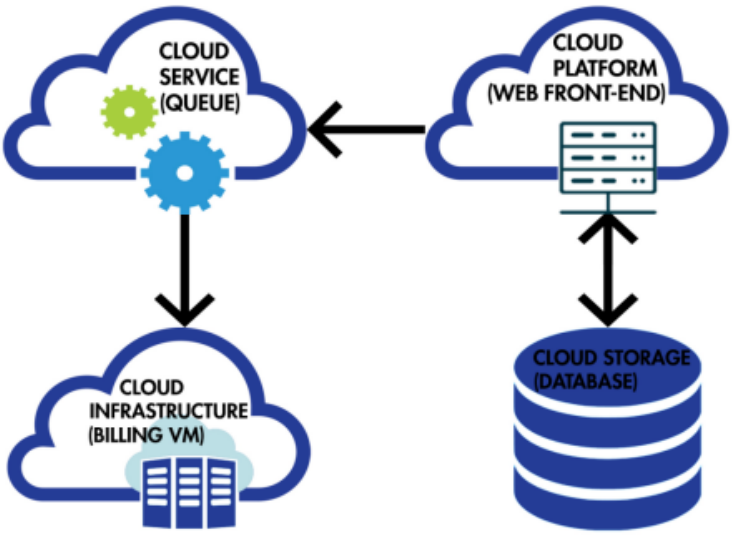


Figure 1.4 Cloud Computing Architecture

The architectural vision of cloud computing is shown in Figure 1.4. The cloud architecture consists of multiple components and sub-components. The parts include Virtual Machines and cloud services and backend systems such as servers, storage, and cloud-based distribution through the internet, the intranet and Intercloud networks.

- **Cloud client platforms**: A typical cloud architecture is a front-end network known as customers or cloud customers. These customers might be agents, thick or fat clients, thin clients, nil clients, laptops and smart devices. These client platforms communicate with the cloud data storage through a middleware program, web browser or virtual session.

- **Cloud storage**: Online storage of the network where various customers store and view data. Normally, cloud storage is used under various frameworks, including public, private, cloud-based or hybrid cloud configuration.

1.1.1. SERVERLESS COMPUTING

The Services-Oriented Architecture (SOA) has been a key discipline for addressing the difference between IT and business services. In recent decades, companies have created SOA and distributed networks that cover numerous business areas, including letters of mail, logistics, financing and banking. The key of SOA is to produce standardized communication protocols for individual, self-describing software modules, all of which are usable and accessible through the network. Web services technology has incorporated this idea as a significant implementation in the construction of new web applications.

For years, web applications built on a three-tier architecture have been created, which views an application as three separate layers – presentation, corporate logic and data. The architecture essentially specifies how problems are separated between frontend, data and backend. The principle of decoupling helps developers, without affecting other levels, to make major layer changes and thereby easily maintainable applications. In the meantime, business logic and data layers can be introduced as web servers from the viewpoint of SOA architecture, allowing connectivity using common protocols based on the Internet. Web services have recently published APIs to teach users how to communicate and share data.

A webservice that runs on a physical or virtual machine, is installed and supported by a database server. The web server supports HTTP data transmission, which listens to and returns HTTP queries and replies, as well as HTTP requests. There are specifications for setup and servicing when they are hosted on a physical or virtual machine. In order for a computer to be able to run the web service, a development team has to review and determine machine requirements (such as processing power, memory and storage spaces). The phase involves the acquisition of hardware and software-licenses as well as time and team commitment to set-up whether the server is installed in-house. Such a web server demands routine servicing, such as error detection, rational emergency management planning and updating (the server is either over-loaded or crashed), which requires human resources (both software and hardware components).

Web systems will, on the other hand, be implemented on a computer network, where cloud providers have the infrastructure and computer tools needed to host and execute applications. When this method is used, the team discards questions about hardware modules, but the work of configuring the server and operating system is still persistent and requires human efforts.

Cloud providers recently launched serverless computing systems, also called the Function as a Service (FaaS). For instance, Amazon's computer services, Amazon Web Services (AWS) Lambda, provide the container of an ephemeral feature for the execution of the code of the application. The container is an environment fully configured to work with such source code. The team will then focus on writing back-end codes for corporate logic and installing them on networks without regard to the infrastructure sophistication and maintenance.

In addition, Function as a Service is used on request, and the standard server framework operating behind the program will also be removed. This function code is used on request. This concept helps to create a network services solution, namely a serverless architecture for building, deploying and distributing web services in a cloud environment with a server.

To explain, a disclosure of the serverless architecture does not announce server participation not withstanding its name. Indeed, there are servers that cloud service providers set-up and maintain.

From the programming team point of view, the word serverless is used in which servers and virtual devices do not buy, rent and retain code for the program. Instead, the actual computer energy used to run the code is compensated. In other words, there would be no fee if the codes did not work. The method of serverless architecture thus allows to some extent to minimize operating costs compared to the conventional solution of maintaining a running server continuously.

Contextualizing Serverless Computing

What technological advances have been needed to facilitate serverless computing? Some have suggested that serverless computing technology is just a rebranding of previous offerings and maybe a modest widespread use of cloud products from Platform as a Service. Others might point out that the common web hosting environments in the 1990s provided a lot of computers without servers. They had, for example, a serverless model of multi-tenant scheme, elastic response to varying demand, and a structured Common Gateway Interface (CGI) Feature Invocation API, which enabled the direct deployment of source code in languages of high level, such as Perl or PHP. The Google App Engine, widely rejected by the market a few years ago, has enabled developing companies to install code while leaving other facets of operations to the cloud provider. The App Engine has been rejected by the market. Computing without a server is a major innovation compared to PaaS and other previous versions.

Today, the serverless cloud storage architecture varies in a number of important respects from its predecessors: greater self-scaling, tight isolation, simplicity of the network and support for ecosystems. These are the factors which marked a striking start from what was before the AWS Lambda autoscaling was offered. It has monitored loads with much greater trustworthiness than server self-scaling strategies.

When required, it reacted rapidly and reduced them to zero resources and zero costs in the absence of demand. It paid even more sophisticated costs, offering at a moment, while most self-support systems charge an hour, a minimum bill increase of 100 ms. 8 During a crucial beginning, the consumer paid for the time that their code was executed rather than the money allocated to running its software. This differentiation ensures that the cloud vendor has auto scaled and thus encourages effective resource allocation.

Serverless computing depends on high efficiency and safety insulation, to allow the sharing of multi-tenant equipment. The standard for MD-like isolations for multi-location hardware sharing for cloud functions is new, but VM providers may use elaborate technologies to speed up the development of functional performance environments for several seconds without the server. A “warm pool” of VM instances, only a tenant must be reserved, and an “active pool” of instances that have been used to perform a function previously and are retained to serve for potential invocations is one approach mirrored in the AWS Lambda approach. Resource life cycle management and multi-tenant bin packaging are key technological facilitators in serverless computing to ensure high levels of use. The recent initiatives aim to decrease the overhead of the use of bins, unikernels, library operating systems or language VMs for multi-tenant isolation.

Several other differences have contributed to the success of serverless computing. The serverless computing system can support much wider implementations than PaaS services, which are tightly related to specific uses, by encouraging users to bring their own libraries. Serverless computing works in new data centers and is far larger than old shared network hosting environments.

The serverless model has been popularized by cloud functions (i.e. FaaS). It should nevertheless be acknowledged that the popularity of BaaS services since public clouds began, services such as AWS S3, is a part of their success. These services are, in our opinion, domain-specific, highly optimized serverless computing implementations. Cloud functions are more often represented by serverless computing.

Attractiveness of Serverless Computing

Serverless service helps market expansion by making it easier to program the cloud, which attracts more users and improves current customers' use of cloud technologies. Latest studies have found, for example, that about 24% of serverless users are new to cloud computers, while 30% have serverless computing used by current cloud clients. Furthermore, the short-term, limited memory and statelessness of cloud services enhance computational multiplexing by facilitating unused tools for these activities. Cloud providers may also use machines that are less common because the instance form is accessible by cloud providers — for example, older servers that are less appealing to server cloud users. Both advantages boost revenue from available services.

Customers benefit from greater productivity in programming, and cost efficiency can also be achieved in many cases when the underlying servers are used more often. While serverless computing makes it easier for consumers to do so, Jevons paradoxically implies it would boost cloud usage instead of reducing it as greater reliability increases demand by adding users.

Database consumers prefer serverless computing, as novices can implement features without any cloud architecture awareness, and since professionals can save time for implementation and work on application-specific issues. Serverless consumers can save money because functions are performed only when incidents arise and fine-grained accounting (usually 100 milliseconds today) ensures that they only pay for what you use according to what they book.

Investigators have been drawn to server-less and cloud functions in particular, as it is a modern abstraction of calculation for the general purpose that aims to become the future of cloud computing and as there are numerous possibilities for enhancing existing efficiency and addressing its current limitations.

Limitations of Serverless Computing Platforms

For many working categories, including API service and event stream management, and minimal ETL11, the serverless cloud functions were effectively employed. We tried to make a serverless version of applications that were of concern to us, and we reviewed examples from others, in order to see what barriers hinder users from embracing more general workload. They are not meant to reflect the majority of IT outside of the existing serverless computer ecosystem, they are merely examples chosen to reveal certain typical vulnerabilities that may preclude serverless implementations of many other fascinating applications.

1.2. SERVERLESS COMPUTING SECURITY

Serverless computing reshapes the protection obligations and shifts many from the cloud customer to the cloud vendor without modifying them radically. The dangers of both the device's disintegration and multi-tenant resource sharing must also been taken into consideration by serverless estimations.

- **Scheduling randomization**: The center of hardware-level side-channel or Rowhammer attacks in the cloud is physical co-residency. The opponent should affirm the coexistence of the victim with the same physical host as the first stage of such attacks, rather than arbitrarily targeting the end users. The capacity of the intruder to distinguish running victims can be reduced by the ephemerality of Cloud functions. The probability of co-locating the perpetrator and the victim may be reduced by a randomized, adversarial scheduling algorithm, which makes joint ventures much more complex. However, the physical co-residency may intentionally preclude an investment in order to maximize startup time, use of resources or connectivity.

Fine-grained security contexts: Fine-grained configuration is needed for cloud operations, including access to private keys, storage objects, and even the local temporary resources. Security policies are needed for translating from current server applications and for the delivery of high security APIs for complex cloud use. For example, a cloud feature may have protection rights transferred to another cloud feature or cloud service. Cryptographically secure protection system built on the ability to monitor access may be a natural match for such a distributed model of security. Other problems in the distributed control of authentication primitives are compounded as short-lived keys and certificates are dynamically generated for cloud functions, such as non-equivocation and revocation. In the device, users need, at least as an alternative, more refined safety insulation for each feature.

The challenge for functional sandboxing is to keep a fast initialization period without hiding the running environments in a way that divides them between repetitive function calls. One way to start each feature from a clean state will be locally to the instances. These isolation strategies limit boot time to tens of milliseconds relative to the VM boot time in seconds. Whether these technologies reach safety parameters with conventional VMs remains to be seen and we expect an active field of research and development in the quest for strong separation mechanisms with low start-up overheads.

On the other hand, the control providers and short-term instances will allow for a much quicker patching of serverless computing. One alternative would be to require physical segregation for users seeking protection from co-residency assaults. Cloud

providers could sell consumers a premium alternative to launch functions specifically for physical hosts.

- **Oblivious serverless computing**: Cloud functions may communicate patterns of access and timing information. Data is typically downloaded and cached locally in a batch for server applications. By comparison, because of cloud features are ephemeral and are broadly spread across the cloud, network transmission patterns can leak more sensitive information from a cloud network intruder (e.g. an employee). This security exposure is exacerbated by the propensity to decompose serverless software into many little roles. Though the end user attackers are the main security issue, network patterns may be shielded by overlooking the algorithms. These leads to have a heavy overhead, unfortunately.

1.2.1. SECURITY ISSUES IN SERVERLESS COMPURING:

Serverless has its own difficulties in different ways, as set out below (Dillon et al. 2010.).

- **Security and Privacy:** Data security for any company which is still subject to inspection, is a fundamental part of this. Because of security concerns, companies are hesitant to purchase insurance from the providers [3]. In competition with rapidly moving technologies, they risk losing consumer data and their secrecy is high. The real storage, which adds to the security issues of the companies, is also not revealed. This confidential information is protected by traditional firewalls through data centers (owned by companies) [4]. Service providers in the cloud model ensures the safety of data on which companies can rely blindly on them [5].

- **Data recovery and availability:** Service level arrangements are strictly adhered in all enterprise applications. Operational teams play an important role in service level agreement administration and program runtime governance. Below are some of the activities carried out in the manufacturing area by the technical staff.

- **Lack of standards:** Cloud platforms have documented the interfaces, because of the requirements of these interfaces are not limited, the cloud is typically interoperable. The Open Grid Forum develops the Open Cloud Computing Interface to address this problem and the Open Cloud Consortium works on the principles and policies of cloud computing with ideas from different angles. The results of such groups demand further guidelines, but it is not certain if the services are designed to meet people's requirements by the deployment of suitable interfaces. But staying up to date with the new developments will increase the value of the results.

- **Interoperability:** Software applications should be able to use other platform services. This is possible through web services. However, it is very complex to develop such web services.

- **Computing performance:** In order to deliver cloud-intensive computer services, high network capacity is needed, which means greater costs. When achieved with a lower bandwidth, it does not fulfil the required performance of an application.

- **Portability:** This is another problem in the cloud world that allows programs to move from one cloud vendor to the another. Vendor lock-in problems should not be present. Since, the usage of many common languages that cloud services use on websites, this has not yet been made possible.

- **Destruction of Data:** When no further data is required, the data must be entirely deleted. Because of physical storage characteristics, the lost data is still available and may be restored or retrieved. This obviously results in classified data being disclosed to unauthorized cloud parties.

- **Group Key Agreement:** Multiple network nodes participating in a common group, sharing a common cryptographic key. This is an area of active research as restrictions on both computational complexity and network utilization are frequently imposed. It must be enforced that perfect backward and forward key secrecy be maintained to prevent any decryption of any messages received while not a member of the group.

- **Confidentiality:** It is computationally infeasible for the intruder to determine the message other than the receiver. This can be achieved through the use of cryptography. The main goal of cryptography is to keep data scheme and utmost privacy of the data. In two-party communications a series of options exist. For symmetric cipher systems, a shared common key is fed into an encryption scheme. This key is agreed upon using a key agreement protocol, much like Diffie-Hellman. For asymmetric cipher systems, a set of keypairs must exist and be known. A sender would then encrypt a message with the receiver's public key, and the receiver would then decrypt with their private key. This requires no key agreement, but does require an up-to-date key database that must be known and trusted to be secure and safe from tampering. In multi-party communications, as in collaborative broadcast-based groups, encryption serves as a membership border. By the nature of broadcast, all nodes within a sender's broadcast range receive the message, but with encryption, only those nodes with knowledge of the encryption key will be able to understand the contents.

- **Integrity Verification:** Knowing and being guaranteed that a received message has not been tampered with or altered in any way. This is generally done with the employment of hashing. A message of n length is represented by a constant-length bit string of k bits. Any change in the message results in a dramatically different hashcode, and any alteration of the hash will clearly not match the data. Using just hash code values does not prevent a third party from modifying the payload and injecting a new and valid code.

- **Authentication:** Knowing and being guaranteed that a received message from sender "S" was in fact sent by 'S'. This functionality is combined with integrity verification, and provided through the use of digital signatures, and, in a more connected world, supported by digital certificates and the Public Key Infrastructure (PKI). Signatures are based on public and private keypairs. A message would be signed with the sender's private key, and later verifed with the sender's public key. This provides proof a particular sender sent any given message. Signing includes a mixture of private key and a hash of the message. This verifcation may be done by any party privacy to the signature and the signer's public key.

1.3. MOTIVATION

The emphasis of this research is to address on data security and data protection issues, where a stringent privacy-preserving solutions are needed. The main focus of this study to provide and safegaurd medical research records, which calls for especially stringent solutions to maintain privacy. For e.g. a researcher who aims to gain insight into the human body and disease processes through big data analytics and the serverless use of ICT. However, the legal and regulatory issues relating to data rights must be taken into account when using data without server. This records ought to be transparently processed to avoid the identities of the persons who "own" the data. Serverless solutions would also properly protect data privacy. Meanwhile, much of the recent regulations on privacy hinders medical institutions from using serverless services - partly because of the manner in which knowledge management functions are generally established for medical information and partly because of limitations enforced by current medical data management laws.

Serverless machines have taken up the multi-tenancy, lack of power and trusts with many security concerns. Serverless providers have virtualization and required infrastructure to share among multiple users. Multi-Tenancy refers to the multiple independent users or organizations that are sharing a physical hardware and the services of virtualization.

Another possible security violation that may arise is lack of control when user data, software and services are hosted in serverless provider premises. Loss of control since the users have no clear control over their data, serverless providers may conduct data mining on users' data, contributing to security concerns. Furthermore, the users cannot be assured that when they erase their data anywhere when serverless provider's backup data at multiple data centers. This will contribute to the unused data being misused. If the user lacks control on the data, he or she finds the serverless provider a black box where he or she will not manage the resources directly in a straightforward manner.

The protection concerns of serverless computing are responsible for a variety of privacy issues, as privacy is a dynamic subject with varying meanings based on backgrounds, cultures and societies. Furthermore, privacy and confidentiality are two different concerns, while protection is normally required to ensure privacy.

1.4. PROBLEM DEFINITION

The serverless works simultaneously, along with the clients to access their files that are stored in the set of serverless, anytime. To prevent the data theft, data is being processed before they are stored in serverless by employing coding, replicating data in various servers to tolerate and reconstruct the lost data. The owner of the data stored in serverless verifies the data integrity in some interval of time without any local copy of it. If the client is busy and could not verify the data in serverless, they hire a third party to do the check without privacy breach. The client sometime do block verification for practical application when it necessary.

When the data are dynamic, i.e., when the data gets updated or appended or deleted frequently, the user verify the data integrity at certain regular time interval. When the data outsourced are very sensitive and large like, military, banking, medical history there are many applications that are envisioned to adopt for storage system.

The control of the data stored is completely given to the serverless service Provider by the clients as they do not maintain a local copy. The adversary can edit or erase the data or induce a virus which corrupt the data storage server or eavesdrop. There is a high risk to vulnerable attacks or system failures which bring heavy loss in the business enterprise as they store the data outside their organization. These attacks may lead the client to loss their authority and access to their own data. Hence, it is important to implement an efficient protocol to ensure authentication, integrity and availability of the client data in serverless with least overhead.

From the perspective of the client, the threats are captured all available intruders towards the data originality because the data store do no reside in the local site of the user but in the serverless service provider domain. Both internal and external threats have their benefits in stealing or corrupting the data for financial advantage. The internal threats are the trusted parties who hired to secure the data stored in the serverless but for they are self-centered serverless service provider who is malicious. The internal threats allocate the data storage place to another client as the data stored is rarely used and hide it from the data owner. Sometimes they are not trustworthy as they hide the data loss incidents due to various reasons like Byzantine errors, from the user or the client.

The External or outside attacks are not within the serverless domain of the service providers. The unauthorized users who are capable of intrude the server and access the data without the knowledge of the client and serverless service providers. The attacker may corrupt or leak the secure data of private user example sensitive military ideas, banking credentials etc. Some of the external adversaries includes:

- Masquerade: The attacker pretends to be the authorized person of the data.
- Replay: The Previous authentication or any information is regenerated by the server for the integrity verification.
- Eavesdropping: Continuous monitoring of the communication and surveillance the communications channels for leaking the information.
- Malicious Instructions: these are programmed instruction to corrupt the information stored in the server.
- Modification: Editing the content of a data block or sometimes deleting the whole file.

1.5. OBJECTIVE

The main objective is to elaborate the basic concepts of serverless computing which define the nature of the problem definition and motivate to extract the better solution of the data security in serverless storage.

The serverless service provider possess a pool of resource which can be outsourced as it is referred as a virtualization. Each consumer is assigned and release the resource on their demand, where the customer is unaware of resources location like database, CPU etc, but they are allowed to limit their physical location which satisfy the legal agreement.

- On demand self-service: Without human interaction the consumer may be provided with requested resource from the serverless service provider such as CPU time, software rent, storage and so forth. The above factors reduce the cost as it is automated for overcoming personnel overheads.

- Rapid Elasticity: The resources are increased as its capabilities for request service and release it to the customers. Procurement time can be decreased for new computing methods by automated processes.
- Broad network access: Though internet only the serverless services are accessed by personal computer interfaces, complex device, smart phone, laptop which consumer desired
- Measured service: Each service provided is measured for checking utility of computation. It provides resource efficiency feed to the serverless service provider to fix their charges for pay and use strategies.

1.6. DESIGN GOALS

The enhanced dependability and security of data that resides in serverless with the risk of the adversaries threats of data confidentiality and authentication of data and the serverless users, the proposed protocols build a proficient methods to secure the dynamic data and its operations. The following are efficient goals to be achieved.

- **Less storage cost:** The additional storage charges used for auditing should be minimal for both serverless and the auditing side.
- **Minimum computation flaws:** All the computation process like proof generation, verification, and initialization should be precise and efficient.
- **Low communication overhead:** The verification by client and response generated by the server should be of minimum cost and less complexity.
- **Availability:** The data storage should be resistant against malicious attacks, attack on servers and byzantine failure where the data can be reconstructed or copied from the multiple copies stored.
- **Confidentiality:** No unauthorized parties can access the data stored in serverless, the protected data are sealed and cannot be accessed without proper authentication process.
- **Integrity:** The consistency of data originality should be maintained by protecting it from authorized data accessing and spotting the corrupting server.

1.7. PROPOSED WORK

In the proposed research work, Security framework is broken into two phases – privacy and message authentication. Privacy is supported through encryption, with authentication through digital signatures.

Two important aspects about groups and group security must be highlighted. The first is that smaller devices cannot compare computationally to larger computing devices. As a result, these smaller devices run a risk of cryptographic compromise. The nature of signing a message is a partial revealing of sending a party's private key. In the meantime, the entire private key is revealed, and determined third party could begin forging valid message signatures.

The second observation is that the keys and signatures used in this method are very short-lived. Public keys used in key agreement and signature verification are valid only as long as the lifetime of the group or the duration of a members' affiliation with the group.

Message integrity in the strict sense is ensuring data received has not been modified. A naïve approach would be simple checksums, but a drawback to this technique is that a third party may modify the data and insert a modified, and correct, integrity checksum. Clearly this approach will not be safe and secure.

Message Authentication Codes (MAC's), or more specifically Hashed-MAC's (HMAC's) are also inappropriate for this environment. These constructs provide data integrity validation, but do not provide message authentication, as the authenticating bytes are generated and verified using the same cryptographic key. Message authentication codes are useful for quickly determining if a file on a computer has changed, but little more.

Traditional algorithms for message authentication rely on digital signatures, generally accompanied by a digital certificate chain to prove authenticity. Digital signatures provide the same unique hash value as MAC's, but include verifiable data from a private key value. This allows a receiver (who, it must be assumed, possesses the associated public key) to verify both the identity of the sender and the message data.

While the dynamic network system will also utilize digital signatures, one critical difference exists with a more connected computing paradigm. In this method, there is

no way of verifying a chain of trust as presented in the certificate. This means, technically, we really do not know who the sender of a message is, but we do know that the message was not modified in transport. We can associate a message with a device or a user, but have very little proof of user authenticity.

Additionally, signature and key agreement keys are separate entities. This is done for algorithmic and mathematical integrity rather than security purposes. The unique strength of elliptic curve keys is presented in this research work – using the same keys for generating the common group encryption key as for signing messages.

The private key is used to sign a message, whereas the public key, distributed as part of the key tree to the entire group, is used for signature verification. Other research does not explore this application of dual-role keys. This technique provides us with several desirable factors.

- First, we have one key to distribute per node instead of one key for encryption and one key for signature verification.
- Second, when a node serves as a group for joining or leaving a member, their keypair is regenerated. This enforces both forward and backward secrecy from an encryption point of view, but also updates the signature keys are our dynamic and temporal signature system.

The main contribution of the research work is given below:

- In the first part of the research work, an identity based cryptosystem for secured authentication is designed that uses ECC model for securing the data in serverless environment. The segregation of initialization phase and authentication phase enables the ECC model to improve the security and authentication of data in serverless environment.
- In the second part of the research work, an attribute based encryption is designed under different security requirements that includes, data confidentiality, collision resistance, attack resistance, non-accessibility of sensitive information before release time and delete data after expiration time. Considering all these parameters, the encryption model offers improved security of data that gets traversed or stored in the serverless environment.

- In the third part of the research work, a filter based security approach is developed that undergoes three different mechanism including encrypted fuzzy based filter with link reliability estimation, VM reliability estimation and residual energy factor estimation. Secondly, it includes the adoption of signature verification scheme involving encryption and decryption mechanism.

1.8. ORGANISATION OF THE THESIS

Chapter 1 discuss the cloud computing overview and the serverless computing overview. It provides the details of serverless computing security and its related issues. Further, the chapter presents the motivation, and contribution of the work.

Chapter 2 provides the related works on security models in the field of serverless platform.

Chapter 3 presents the proposed identity-based cryptosystems for secured authentication in serverless computing model. It also s the performance of the proposed identity based cryptosystems with other methods.

Chapter 4 presents the proposed attribute based cryptosystems for secured authentication in serverless computing model. It also provides the comparison of the proposed attribute based cryptosystems over other methods.

Chapter 5 presents the proposed Filter based approach for secured authentication in serverless computing model. It also provides the performance of the proposed Filter based approach than other methods.

Chapter 6 discusses the performance of the three models with various performance metrics.

Chapter 7 provides the conclusion and possible directions for future research in serverless computing.

1.9. SUMMARY

The enhanced dependability and security of data resides in serverless with the risk of the adversaries who threats of data confidentiality and authentication of data and the serverless users, the proposed protocols build a proficient method to secure the dynamic data and its operations. The following chapter provides the details of the proposed model to secure the data and proper authentication of data in serverless systems.

CHAPTER -II

LITERATURE REVIEW

CHAPTER 2

LITERATURE REVIEW

2.1. INTRODUCTION

This chapter includes extensive background on past studies on data deduction procedures, process of preservation of privacy, self-destruction approaches and serverless user revocation mechanisms. Different methods are briefly examined in relation to the research work. This chapter included a concise overview of all prior studies on deduplication of data, data protection and the processing of user revocations in any computer system without a server. This thesis precisely described the complexities and problems of serverless computing. This research work proposes a better deduplication of the false fire, privacy and dynamic self-destruction updating for serverless data, in order to balance the protection and efficiency strategies in the serverless world.

2.2. RELATED TO SERVERLESS COMPUTING

The serverless data storage project suggested a private deduplication protocol (Ng et al. 2012). A private data deduplication protocol was initially used to allow a client keeping private information which demonstrates the server's summary string. This strategy has been discovered without disclosing any detail to the server to properly manage the description. The security of private data deduction protocols was formulated in the two-party calculations based on simulation. A construction of private deduplication protocols was subsequently presented and analysed based on the traditional cryptographic assumptions. But the numerical complexity of the machine increased. This has been discovered.

A novel coalescing algorithm for serverless storage systems [7] has been suggested for the following year. This algorithm estimates the maximum and minimum number of subpieces to be combined into a subpiece. This algorithm's main objective was to concentrate not just on the maximum amount of chunks to be merged into a size, but also to avoid the lance process when coalescing chunks. On that basis, a new parameter

was proposed, called the low predicted subchunk scale. This new parameter has been used to restrict the number of subpieces, in order to reduce excessive coalescing costs. The deduplication period was nevertheless high. The time needed was high.

In order to boost serverless store performance and preserve redundancy for defect tolerance, a dynamic data deduplication [8] was proposed. Centered on the customer side deduction with complete file hacking process, this framework was designed. Here the Redundancy Manager calculates the best number of copies for the file, according to the number of references and the appropriate QoS amount, if deduplication is found in the system. This copy numbers of files is fundamentally dynamically changed at QoS level. However, the machine assessment was lagging behind in its availability and efficiency.

Enhanced data deductions from nature-inspired data are an innovative framework that is proposed for effective serverless control of storage [9]. This technique involved text matching algorithms for deduplication using the Sequence Matching algorithm (SM) and Levenshtein algorithm, GP. For text comparison and for the detection of the next player, the SM and Levenshtein algorithm were used. This approach allowed users to efficiently use the space offered by serverless providers, thus reducing heavy charges for resource use. Meanwhile, though, the time it took to match the text was relatively strong.

A reliable multi-server-aided data deduction was suggested [10], based entirely on a decentralized blind-signature threshold. The collusion assault between serverless servers and multiple servers successfully withstood this tactic. Each consumer has created this solution by interacting with many key servers to create a convergent key. This prevented any partial key server from gaining knowledge of the hidden key spread across all major servers. However, it was found that the time required to implement this method was high.

Payment-based compensation mechanisms for safe serverless deduplication that enhances data management performance [11] were suggested. This solution included a server-support network protocol that guarantees the validity of the deduplication rate to provide confidentiality protection through Convergent Encryption (CE). In addition, the price function was specified based on the objective of the deduplication. This studied per-bit customer billing and the data storage utilities to service providers with

serverless storage (S-SSP). The overhead storage has been found to be high in the system, despite all these advantages.

A method for stable serverless infrastructure was proposed called allowed deduplication [12]. This strategy has implemented a protected permitted deductible using the token generation mechanism. The serverless service and storage providers have also been able to use the strategy of data deduplication without having access either to user plaintext or decrypted data. However, the efficiency of this method has not been well analyzed. In the serverless storage on the basis of FP-tree, a data deduplication approach was proposed [13]. This solution was proposed to increase data storage performance, which also increased functionality for read and write. At first, the FP-tree algorithm was used to store vast amounts of data and this was shown with computational derivations. A variation of the FP-tree algorithm and data server-free location system was then developed to efficiently handle storage using the serverless server-free backup technique. However, because the input data became too big to process, the test device was analyzed to be unreliable.

In order to improve the serverless platform availability and stability, it was recommended a Deduplication-Assisted Serverless Primary Storage System (DAC) [14]. This solution incorporated replication and erasure coding schemes that replaced redundant server-free infrastructure data blocks, which properly spread data through many different serverless storage providers. The duplication system has retained heavily referencing data blocks, while the other blocks have been saved with the erasure code scheme. This exploited the value of duplication and erasure code schemes by taking advantage of their comparison features. However, for improving safety on the non-server storage systems, DAC demanded reliability analysis and data encryption.

A verifiable data deduplication scheme was suggested [15] that clarified the issue of the validation of the serverless image deduplication. Usually, a serverless server was used to check the accuracy of the data deductibles, where the fingerprint was calculated for its hash value for any encrypted picture submitted by the consumer. These fingerprints were sent to a total of two serverless servers for the verification of duplicates: the Storage and Verification server. In case of no deduplication from servers, the user transmitted the data. The users abandoned the concept of uploading data when fingerprints were repeatedly detected. The findings were inconsistent with

fingerprints on a server alone and made at least one server invalid. In addition, the study of safety and efficiency has been shown to minimize storage space by exploring the deduction in block levels of the picture.

For serverless storage improvement and its functionality a stable replication was proposed of encrypted data called Serverless Deduplication [16]. This solution was suggested with the aim of simultaneously maintaining block deduplication and privacy. Moreover, because the necessity for block deduplication posed a problem about key management, a new aspect was added to apply the core management process for each block and the actual deduplication process. Nevertheless, further protection features such as recoverability, data integrity control and encrypted data search were essential.

Anonymous encrypted data have been proposed as a proof of ownership for serverless storage [17]. This solution introduced a number of features to use separate client-level encryption keys, proof of ownership and customer connected data files on the blind server. Clients used chosen hidden keys to encrypt themselves. Deduction here did not restrict the choosing of the encryption key for clients. Any customer who demanded evidence of ownership of a file must have the whole file. Moreover, the relationship between clients and their files from the serverless server was obstructed using an encrypted channel. The serverless cloud then provided the data owners with a digital credential to retrieve their data files as needed. However, there has been a high calculation time for storage and retrieval.

For serverless infrastructure, a reliable data deduplication approach is suggested [18]. In this context, a new way to safeguard the privacy and confidentiality of information was adopted. This approach used an effective deduplication algorithm for the division into smaller units of a given file. The users who have merged a stable hash function and a block encoding algorithm then encrypted those units. For such hash values, users have built an index tree that has been encrypted by means of an asymmetric search engine. The index trees enabled the service provider to check the index and to return the units properly. The serverless storage providers must not be trusted when managing user data. Analysis of this method, however, was not seen.

The Virtual Machine (VM) was suggested to provide a deduplication-based Energy Efficiency Storage System (EEVS). All VM image files have initially been studied using general operating systems. Based on the study, multiple redundant data blocks

were found to provide additional VM storage energy. Thus, the proposed solution was intended to reduce these redundant data without interruptions in operation through an online deduplication process because traditional deduplication technology has been used for offline backup. An EEVS was developed using the previous serverless platform according to the device architecture. For a certain range of VMs with restricted deduction resources, a deductible selection algorithm was developed so as to minimize storage energy usage. The efficiency of this strategy has, however, been found to be less efficient because the algorithm has taken more time.

For serverless storage to remove duplicated chip text and to improve privacy security, encrypted data deduplication [19] was suggested. This mechanism is used to store all the data in the cipher system, which has a search block and a conversion block. The serverless storage server recognized duplicate cipher structures and translated enabling blocks. The cipher blocks were reported to the same cipher by any data owner of the duplicate structures corresponding to the same plaintext. So only one copy of the replicated cipher block was saved on the serverless disk server to prevent unused storage space. The computer time of this solution was heavy, compared with other mechanisms, despite all its advantages.

For the purposes of data deduplication in serverless storage, a new content-defined chunking algorithm (CDC) [20] was proposed. This technique implemented a high-performance, hashless-chunking, Rapid Asymmetrical Maximum process (RAM). Here, the bytes values were used by RAM for the declaration of the cut points instead of hashes. This algorithm was also used to determine the maximum byte referenced by a fixed window and a variable window. The full byte in the chunk and at the edge of the chunk was used. This setting was used for the RAM's retention of the CDC property to make less contrast. Despite all its benefits, it was necessary to increase the accuracy of identifying duplicate files.

Awareness in application for personal storage serverless backup facilities, the locally-global root deduction known as ALG-Dedupe [21] has been suggested. The key path of this system was to increase the effectiveness of the deduction of data through the exploitation of application consciousness that combined locally and globally. This sparked a strong balance between serverless storage capabilities that save significantly less time. In addition, the utility of various deduplication schemes on one platform were

calculated with new metric bytes which have been saved per second value. The suggested scheme was also used to reduce the computing overhead and maximize the deduplication efficiency with an intelligent data chunking approach and adaptive use of hazh functions. The application-aware index structure has been used to independently eliminate consistency, and has optimised the viewing output by concurrently separating a central indicate into various separate small indices. Furthermore, a customer-side data aggregation approach was proposed which enhanced the reliability of data transfer. This has been entirely focused on the combination of smaller data packets for serverless storage in larger ones. This scheme, however, would not promote safe deduplication where protection is not investigated, which is a big concern.

For stable permissible deduplication, a hybrid serverless solution [22] was suggested. The approved data deduction issue was initially resolved and then the user's differential rights were considered in double checks, in addition to the data itself. Several new deduplication constructions were also proposed to support the approved duplicate search in a hybrid serverless architecture. This technique only permitted the users to search for files with restricted rights duplicately. In addition, an improved scheme to enable enhanced protection that encrypted the file with differently privileged keys has been implemented.

According to the literature analysis, the deficiencies identified in the existing system are high computer time, lower performance, and availability, a lack of coding standards for ensuring serverless storage security, a compromise on data integrity at high computer costs, and unauthorized access without limiting access to data with the application and the data film. This underlines the need for new studies to tackle the problems described above.

2.3. RELATED TO SERVERLESS SECURITY

A revocable key aggregate [23] cryptosystem was suggested for the exchange of information on an essentially serverless sub-set-cover network. The proposal had the core aggregate features, which made key management considerably easier for the customer and revoked access allowances with versatile and efficient access regulation. Whenever the account cancellation has begun, the serverless server has been able to upgrade its ciphertext such that revoked users have access to a new code which is not

compulsory for unrevoked users. A testing procedure was also used to validate the modified chip text in order to ensure that the user revokes are carried out correctly. This strategy was therefore considered only for the construction of a CPA protection system and the expansion of the system found the total number of users to be inflexible.

A crypto-imposed access control system [24] for untrusted serverless storage has been suggested, with a customizable user cancelation process. This framework supports a refined framework for controlling access and allows scalable repeals without data migration to invalid accounts, who mistakenly depend on serverless providers. The framework employed the encryption attribute-based technology that enabled users to identify readable human access policies for data in serverless storage to complicated access structures. Moreover, this system provided a modular revocation system, revoking invalid users in two ways: one which modified the revoked user's list explicitly and the other, which revoked the list implicitly on the basis of an epoch clock. Based on the system requirements, the system administrator has opted for one of these choices. This strategy enabled the authorized users to update the encrypted data and enabled users to validate if the authorized user had updated the data correctly. The framework needed a better updating process despite its benefits, because the calculation time for decryption was high.

For the serverless setting, a new effective Access Management Scheme Attribute Based Revocation (ABR) [25] was suggested. The user cancelation was assisted by an effective fine grains access management system. This method abused the power of the technique for the hidden exchange of community administrators, increasing both attribute and user level immediate and effective revocation. The advantage of fine seed access offered by the CP-ABE technique is totally beneficial. The benefits of this scheme were many, one of them was no machine re-key operation. With the immediate revoking of users, the allocation of the revoking technique to the secret sharing process has modified the secrecy of attribute classes so that only approved people can find out the new secret. However, the technique needed to improve the structure further to achieve improved results.

For attribute-based encryption in serverless storage, a dynamic user revoke and key refreshing technology [26] is suggested. This technique uses the CP-ABE implemented for serverless storage inside a data-owner-centric system to use a Dynamic User

Revocation & Key Refreshing (DURKR) model. To address problems of scalability in current revocation programs, DURKR implemented a proxy re-encryption method. DURKR was one of the programs to first refresh the CP-ABE approach with a dynamic system. It was also planned to be a generalized model that operated with the number of CP-ABE systems in use. This approach, on the other hand, necessitated more computational and coordination time research and development.

The user revocation and update of attributes for serverless applications has been suggested for a serverless access management system [27]. In order to promote account revocability and modifying attributes, this method uses access control (CP-ABE). A secure identity-based revocation solution has been developed to solve ABE-based account revocability problems. In addition, the revocation functions had little impact on the attributes of the code because the names of the revoked users were integrated into the ciphertext during the encryption phases. A powerful upgrade procedure was used to overcome problems with the attribute update because the separate update (key) information for each device user was configured to quickly update the hidden key and the ciphertext. However, the reliability of the user retrieval and upgrade access control scheme is poor, as it is ideal for serverless platforms that use and run a large amount of data.

For a safe serverless world, a fine-grained access control and [28] has been suggested. This method safeguarded the user files, which could offer the data on the serverless fine-grained access control. In order to do so, simple cryptographic systems were used alongside μ-degree bivariate and symmetric polynomials. Furthermore, it employed a modular user control cancelation system that increased the cancelation of interpolation, without changes to other users' shares. However, the coordination overhead was high, so polynomials were used to distinguish data and other operations.

For serverless hierarchical community operations, a powerful user revocation method was proposed [29]. Dynamic encryption methods for broadcast were employed in this technique, using the group signature. The overall Security algorithm for the exchange and distribution of information in an anonymous serverless system was used to create Elliptic Curve Cryptography (ECC). This regime has been determined to be autonomous of various revoked consumers of overhead computing and encryption processing costs. As a result, this method necessitated the use of an appropriate key

management technique for the removal of private keys from any party on the serverless.

In addition to the user cancellation which protects and selectively accesses data in a serverless platform, temporary access control technique [30] has been suggested. This method was used primarily to encrypt and store data in serverless devices in order to be decrypted in the order provided only by approved users. Decryption has been partially outsourced to a proxy server which reduces the system's computer load. This made the system for devices with reduced computing capacity more comfortable and secure. This solution requires an optimization algorithm to increase the efficiency of the user revocation process despite its efficient operation capability.

2.4. RELATED TO IDENTITY BASED CRYPTOSYSTEM

Serverless based deduplication and self-data destruction [31] were proposed for enhancing the security of data stored in an encrypted form on the serverless. Duplication was simultaneously checked before storing the data in the serverless so that the space is managed efficiently with minimal billing to the users from the Serverless Service Provider (SSP). Access permission mechanism was also utilized in which users had decryption and deduplication access permissions. The life span of each data item was provided every time a data gets stored on the serverless. Moreover, self-data destruction algorithm was utilized that automatically removed the data on the serverless with the least life span. However, the performance of this approach was not analyzed appropriately.

An improved data self-destruction [32] technique was proposed for protecting the data privacy on the serverless. In this method, a new scheme named Safe Vanish was proposed that prevented hopping attack. This was achieved by extending the length of the key shares which in turn improved the attack cost substantially. This enhanced the performance of the Shamir Secret Sharing algorithm in the original vanish system. An improved approach against sniffing attack was proposed that employed the public key cryptosystem to protect against any sniffing operation. However, several limitations were noticed in this technique. This approach was found to only improve the cost of hopping attack, but could not stop it completely. Another an important issue was availability, when the key shares were extended to a specific extent to be handled easily.

A controllable data self-destruction system named Serverless Sky [33] was proposed for handling untrusted serverless storage networks. This approach had the ability to enforce the security of user privacy over untrusted serverless in a controllable manner. It exploited a key control mechanism based on Attribute-Based Encryption (ABE) and with the help of active storage networks. These networks allowed the users to control the life- cycle and the access control policies of the private data. The integrity of the data was ensured by using HMAC (Hash-based Message Authentication Code) to operate under untrusted environments. However, this technique was not sure to be scalable if the data applications were of varied dimensions. In spite of all its security measures, technique lagged behind due to fraudulent messages over digital signatures.

An ABE-based secure document self-destruction scheme (ADS) [34] was proposed for self-destruction of documents on the serverless securely. This technique employed ABE algorithm on a global-scale, Distributed Hash Table (DHT) network that was totally decentralized. The ADS scheme protected the privacy and security of the archived and saved personal documents. This made all the document copies unreadable triggering automatic destruction on the constraint of the user-specified time. Moreover, the ADS scheme provided a flexible access control to the authorized users among social groups. However, the use of DHTs did not guarantee on data integrity and consistency.

A secure self-destruction of shared data [35] methodology was proposed for the multi-Serverless IoT environment. This technique employed encryption of data by access strategy that allowed the owner of data transmitted to the serverless to grant access to only a specific set of captured data fields of the users with minimum number of keys. In addition, this approach restricted the data owner from accessing the data and keys only till a certain period after which it was destroyed completely to be unavailable for any modification. However, in this approach, the performance effectiveness the technique was not analyzed. Data sharing and self-destruction scheme with help of Key-Policy ABE with Time-Specified attributes known as KP-TSABE [36], [37] was proposed for the serverless environment. In this approach, each cipher text was labelled with a time period where the private keys were associated with a time instant. The cipher text was decrypted only if the attributes time instant was available in the allowed period of time where the attributes associated with the appropriate cipher texts satisfied the key's access structure. In spite of all its advantages this approach shared a file to only one user at a time and the computational complexity was found to be high.

Self-destruction model [38] was proposed for protecting the data in serverless storage based on the Data Storage Center (DSC). This approach was mainly focused on the Key Generation Center (KGC), an independent system on serverless used for generating unique key for each file in the process of encryption. The attribute values were managed by the data center that was employed for downloading the data from the serverless storage. Self-destruction methodology also removed the copies of the key generated from KGC and DSC appropriately. However, uploading and downloading speed of files was found to be low in the system.

A Secure Ciphertext Self-Destruction scheme named SCSD [39] was proposed with the ABE scheme in the serverless. This scheme, employed DHT networks that stored all the sensitive data in the encrypted form with an access key. This also stored the ciphertext shares along with the attribute shares for the data in the network. On the other hand, the remaining sensitive data ciphertext and the shares of access key ciphertext were integrated to an encapsulated self-Destruction Object (EDO) on the serverless. Every node in the DHT networks automatically discarded the ciphertext shares and the attribute shares when the personal data expired. This made the ciphertext and the access key unrecoverable by any illegitimate users in the serverless. However, this approach requires further improvement on security.

A self-destruction scheme known as SEDAS [40] was proposed for protecting the data privacy in serverless storage as a service model. This proposed approach had two major parts namely, secret key part and survival time part. The Secret key part was used for generating a pair of keys through the Shamir secret sharing algorithm whereas the survival time part was used for specifying the time limit for each key. This strategy self-destructed the keys after a user specified time which reduced the communication overhead and network delay. However, the key length specified in this approach was found to be less and required to be increased for providing more data privacy to the users in the serverless architecture.

2.5. RELATED TO ATTRIBUTE BASED ENCRYPTION

A new approach [41] was proposed according to the private matching and min-attribute generalization that solved the problem of privacy preserving in the serverless. The major objective of this approach was to employ the private matching technology that intersected the user's data and the datasets of the service providers without accessing

each other's data. Moreover, the problem of privacy indexing was found to be improved in this approach which removed various privacy indexing problems in the serverless. However, the performance metrics of this approach was not analyzed and hence would lead to unsure functionality of the technique in any system.

Privacy-preserving cross-user source-based data deduplication [42] is a technique that was proposed in the serverless storage to dramatically enhance the security. However, the effectiveness of this approach was found to be poor as it failed to address both privacy and data deduplication simultaneously. A three-tier cross-domain structure [43] was introduced that employed an Efficient and Privacy- preserving Big Data Deduplication strategy known as EPCDD in the serverless storage. This proposed approach resisted brute-force attacks that preserved the privacy and improved the data availability. Additionally, accountability that determined the identical plaintexts of the encrypted messages was employed for assuring better privacy. However, the time complexity of duplicate search was not found to be improved which required additional jobs to be carried whenever the data size increased gradually.

An efficient yet secure scheme [44] was proposed that employed searching of encrypted serverless data on recovering the misspellings and typographical errors that existed frequently in terms of search request and source data. This methodology was achieved using a metric space that constructed a tree-based index. This allowed retrieval of data with only the relevant entries with minimum number of distance evaluations. String embedding techniques were utilized for refining the relevant entries efficiently and securely. This index construction was useful for maintaining the privacy of the keyword trapdoors as well as the stored data. A simple cryptographic primitive was designed that protected the embedding vectors when the serverless servers were enabled for measuring the similarity on the encrypted vectors without using any third party. However, the parameters involved in enhancing the security seems to be more in number which would lead to higher complexity and running time of the algorithms.

A retrievable data perturbation method [45] was proposed and utilized in the serverless for privacy-preserving. Initially, an improved random generator was proposed for generating an accurate noise. Then, a perturbation algorithm was introduced for adding noise to the original data. Based on this algorithm, the privacy information was hidden, but the mean and covariance of data which the service providers require remained

unchanged. After that, a retrieval algorithm was proposed which obtained the original data back from the perturbed data. Finally, the retrievable perturbation was combined with the access control process for ensuring only the authorized users to retrieve the original data. However, several limitations were noticed in this approach where the perturbative data was found to be too sensitive to handle and required large number of keys.

A new user-side personalization framework and architecture [46] was proposed that addressed the privacy issues in the serverless. This method contributed an innovative serverless architecture and framework that offered personalized services by protecting users' information. This strategy allowed the users to handle the user-side personalization and data anonymization abided by the privacy laws that widely involved user-side processing with no personal data leakage from the client side. This made the users' more comfortable inside the serverless with good quality of service delivered. A personal data processing agent was employed in the client side through personalization techniques where the queries were sent to the hosts in an anonymous format. The performance effectiveness of this approach was not found to be analyzed properly.

Dynamic data operation with deduplication in privacy-preserving public auditing [14] was proposed for secure serverless storage. This approach enabled data deduplication under three dynamic data operations namely, data modification, data insertion and data deletion. These operations were done on frequent basis by the serverless service providers for improving the storage efficiency that reduced the data volumes. In addition to this, the costs and energy consumption for running large serverless storage system was found to be reduced effectively. This also ensured that the data integrity in the serverless storage system was achieved appropriately. In spite of all its functionalities the system lags behind due to its high computational cost.

Privacy-preserving public auditing [47] technique was proposed for secure serverless storage. This approach employed a secure serverless storage system that supported privacy-preserving public auditing for the data on the serverless. A homomorphic linear authenticator and random masking techniques were included that blocked the Third-Party Auditor (TPA) from learning any information about the data content stored on the serverless server. The efficient auditing process not only removed the burden of

serverless users from this tedious and expensive auditing task, but also alleviated the users from the fear of outsourcing the data. This technique required higher auditing time between the batch and individual auditing.

An efficient confidentiality-preserving Proof of Ownership (PoW) strategy [48] was proposed for deduplication in serverless platform. In this approach, a novel PoW scheme known as ce-PoW was proposed which was found to be resilient with Convergent Encryption (CE) to many servers that paved ways for poisoning attacks. This approach was proved to be more secure under bounded leakage option of the system. However, content guessing attacks against low min-entropy files remained an open issue to be addressed.

A new privacy preserving technique [49] was proposed for serverless service user endorsement that used multi-agents in its environment. The requirement for a generic privacy preserving framework was discussed that performed a decisive task in preserving user's confidential data stored in the serverless storage service provider. However, this algorithm required further improvements on considering the policy and authorization strategies in a dynamic real time serverless infrastructure without affecting the performance of serverless computing paradigm.

Privacy-preserving outsourced association rule mining [50] strategy was proposed on vertically partitioned databases in the serverless. In this approach, an efficient homomorphic encryption scheme and a secure comparison scheme were proposed for ensuring the data privacy. Then, a serverless-aided frequent itemset mining solution was introduced that was used for building solutions for association rule mining. These solutions were designed for outsourced databases that allowed multiple data owners for efficiently distributing their data securely without any compromise on the data privacy. However, the communication traffic and storage cost for such operations were found to be too high to be implemented. Privacy-Preserving certificateless Provable Data Possession (PP-CLPDP) scheme [51] was proposed for big data storage in the serverless. The main objective of this approach was to improve the security of the CLPDP scheme and to ensure the privacy protection. However, the computational complexity of this approach was found to be high.

2.6. RELATED TO FILTER BASED APPROACH

The number of challenges is unlimited in the simple deterministic protocol proposed by [52]. Using RSA based homomorphic hash function the integrity of the remote data is verified. It can be demonstrated by possessing a set of data to the client. The system first initializes the user to calculate the hash value of the file by Euler function and send the data to the server. The locally maintained copy is deleted and retains only the hash values which calculated. A random integer is chosen and sends to the server for integrity check. The server responds to the client and sends back to the server. The results are matched and ensure that the message data is secured.

Through challenge –response protocol the verification of the remote data integrity is proposed by [53] called as remote data checking protocol. Random selection of the element g is done, which computes the homomorphic tag for the file to be stored in the server. With the m data and N public key of the RSA, the client sendsm to the server along with the homomorphic tag t. The verification process begins with the client who chooses a random number r and sends a challenge to the storage server. The proof is generated once the server receives the challenge and sends the response. The proof is verified by client, if the data are intact and maintained consistency then it true or else it is corrupted. The disadvantage is that only fixed number challenges can be employed.

The complexity of computation increases in both the system [52],[53] as the server is meant to hold huge amount data and it impossible to cover the entire storage for integrity check.

To cover the whole file for integrity check in the server [54] is proposed a system which works on efficient remote data possession checking in the decisive information infrastructure. The file divided into blocks along with the homomorphic hash function generated by RSA for each block. The Diffie –Hellman key exchange method give a notion for the data possession checking protocol. It is the same process as mention in [52], where the client divided the data into m blocks and store along with homomorphic hash value. The flaw of this proposed system is that the client has to store hash values of each N size bits which occupy large space at the client side which make it complex and audit process slow and does not support public verifiability.

For quick and fast integrity verification scheme for checking the integrity message, the homomorphic tags are employed in all data blocks and batch verification takes place as projected in [55]. In some cases like [54] the hash values are store in the client local storage where it size is linear to the number of fragments. SEC (storage Enforcing Commitment is proposed by [56] for deterministic verification approach. This system also uses the homomorphic tags whose number is more than two times as the data chunks. The indexes of the tags are shifted by choosing a random value which associated with the data chunk. The flaw of this system is that the size of the client's public is bigger than data file.

To detect the modified data files a scheme is proposed by [57] which is based on the tweak able block cipher. This cipher is used to detect the data blocks which are modified without been authorized. Less space is needed if the trusted client data has low resource, then the client have to keep data files less. The whole file is retrieved for the verification process the communication is complex as both linear and files which under the challenge. The disadvantage is it not feasible the verification data possession with the above complication.

A very common method is proposed by [58] in which the prototype checks the integrity of the data in unreliable storage environment. A separate server is used for authenticating purpose which uses to host the verification though the client holds a small space locally. The cryptographic hash value of constant size and authenticated skip list data structure are stored reliable environment in which application is hosted. The system is very transparent and detect the corruption on both of the server even if threat attacks the both the server.

The confidentiality is lacking in all above protocols though it verifies the originality of the data. To address this problem of remote storage integrity and confidentiality a design is proposed by [59]. The design is based on the tree construction using MAC to assure the file system maintain its integrity and confidentiality. It uses the universal – hash based MAC to enable the security proof for the betterment of the performance compared to the Merkle hash tree. The data files are ensured security by file encryption the system provides both confidentiality and integrity.

A secure file system is proposed for insecure network called SiRiUS by [60] which is assumed to be untrusted network storage gives access to read-write cryptographic

controls foe file sharing. There are minimal out of band communication and key management and repealing are quite simple to handle.

Similar to the SiRiUS, there is design which provides security for file sharing over un-trusted servers. A stackable file system proposed by [61] which explains CRUST, which have sharing policies more flexible by maintaining access privileges per user like owner of the file, read- only, read- write authorization. Usage of MAC based signature in the place of either secret signing key or public verification key.

The Plutus is a new secure file system is proposed by [62] which provide strong security even the data file reside in the pool of unworthy servers. The main advantage of Plutus is that all keys distribution is handle in decentralized manner and all data are encrypted before storage. The confidentiality is maintained while their communication between the user and server (RPCs). All the above deterministic protocol discussed assures the integrity and confidentiality and do not support public verifiability.

The protocols which concentrate on the availability of remote data storage are discussed below; a secure and self-organizing storage (P2P) protocol is described by [63]. It ensure integrity availability and confidentiality with public variability at low resource overheads in periodic verification. The scheme is mention in four phases: setup, storage, delegation and verification. Using Elliptic Curve cryptography (ECC) the client encrypts the data files and generates metadata in the setup phase. The client communicates with the storage server at the storage phase and metadata is given to the verifier either the client or outsourced verifier at delegation phase. The owner checks the data storage server for its originality at verification phase. Using challenge – response protocol the process is executed any number of times using the tiny security metadata. The process assure scalability as it enable data redundancy while avoid peer collision and third party is hired to do the verification for the owner.

To verify the originality of information data stored in the serverless servers a DEMC-PDP scheme is proposed by [64]. Many differentiable copies of the file f is generated by the owner and it is encrypted using AES. There is single secret key k which is maintained by the authorized users which is decrypted later. All the copies are attached a verifiable tags generated by BLS to each block in the file [65]. The cheating serverless service providers are prevented by misbehaving by utilizing blocks from various files in the case of MR-PDP [66]. The local copies are deleted once the owner send the file

along with the tag to the service providers. The verification is done on all outsourced data copies by challenging the CSP to make sure the stored data not less than n copies and not corrupted. Using metadata and the response generated by the CSP is verified. These protocols all works on the static data and not efficient for dynamically changing data.

For data dynamics in serverless server a verification scheme is proposed by [67] which preserves the privacy of the remote data checking protocol in the serverless computing. The above supports the dynamic operation and public verifiability mention in [54]. To Public verifiability the client allows their trusted third-party auditor who knows the credential like public keys and performs integrity check for the client. There are vulnerabilities when the credentials are given to the third party. To ensure the data inside the server is safe from internal and external attacks the privacy must be preserved from third party auditor also without interfering in the verification part [67]. The dynamic data which is updated frequently by appending deleting, adding in-between must be intact without any leak or data loss. After each update the metadata also should be changed accordingly which leads to reliable verification process. The updates should not more complex and these protocols do not support data leakage from the internal attacks effectively.

The unfaithful storage server is described as of how handle them by Provable data possession (PDP) scheme [68]. A random set of data blocks are sampled for probabilistic proof generation without retrieving the original file which is stored in serverless server by the client, in this way the input and output cost is reduced by PDP scheme the scheme is divided into two main schemes: Sampling PDP and Efficient PDP (SPDP and EPDP) schemes. These schemes only give assurance to the possession of the sum of the block and do not guarantee each block is challenged for its integrity.

Homomorphic authenticators are used in these schemes; the homomorphic verifiable tags are computed to get a single value from multiple file data block. Before the data are outsourced to the serverless server there are tags computed for each and every blocks of the data file and stored in the server along with the tags. Frequently the client randomly verifies some selected blocks by challenging them. On the reception of the challenge the server generate proof of possession along with the tags and send to the client. The client verifies with the metadata which client maintain locally and give

response convincing the proof withdrawing the original file from the server. Nevertheless, the security proof is not rigorous for the above protocol.

Building a public-key HLA (homomorphic linear authenticator) satisfying homomorphic properties using identification of protocols is done the subsequent work [69]. The conversion of public-key HLA into publicly-verification proof of storage Pos) is shown along with communication overhead irrespective of the length of the file and supporting boundless number of verifications.

The verification on accuracy of the data in online service [70] is done by auditing the data in the server without any explicit knowledge about the data files the above uses only minimum of memory space. First sequences of requests are sent to the data structure online. The adversary views the data structures which are under the control of them. Using the reliable memory the checker performs each operation in the input sequence, the operation in the unreliable data structure also take place according to the input sequence so that the error can be detected by the verifier with higher probability. Using Heap or Binary search trees the checkers discover more for complicated data structure.

Sequel of the [70] the memory checking schemes for large and remote storage at unreliable server is the problem discussed in [71]. The client store randomized fingerprints in the personal computer which is private in order to check the file corruption. There are issues in sub-linear authentication and the setting must be understood for the authentication problem. The encoded file is stored and make sure is not corrupted without accessing the whole file. The online checking is efficient with the one way function but the sub –linear authentication is inefficient and cannot hold for high risk adversary models.

Provable data possession (PDP) is proposed for the outsourced database according to [72]. In the database a small number of tuples are inserted by the client. When a query is sent to the augmented database, there are some probabilities that few tuples are being inserted. Monitoring is done by inserting tuples response and analysis it for the integrity of the system. The client should be aware of all tuples inserted in the outsourced database for analysis. To identify the security breach, the client check for the inserted tuples reply is missing in the response. If all query receives the reply with the inserted tuple, then the probabilistic assurance on query Integrity satisfied.

There are many challenges involved in this task. The client holds a copy of all inserted tuples, in order to know the set of inserted tuples as which had to be returned which require of store locally and query processing is also done locally. In the above approach a deterministic function is used to describe about the inserted data. The encrypted function generates the data which is very difficult for the service providers to differentiate the inserted data and encrypted database. So, they need retain only the definition of the function at client side and not the data. The query process must be secure and does not provide any clue or a little information to adversaries. If the inserted tuples from the original tuple is revealed to the outsider, then the scheme is a failure. To secure the database form the threats a provable scheme is proposed and achieves a high-level safety from adversaries and computational bounded. The integrity verification for join and updates of the database is provided by scheme. Data processing services like search engines, storage, backup system, etc. are application of the above technique. The disadvantage of PDP protocols is the availability of data i.e., the file retrieving is not assured for the outsourced data but provides validation for that.

The data stability, integrity and availability are tackled by [66] Multiple –replica PDP (MR-PDP) scheme which is implemented on the deceitful storage servers by replicated mechanism [73] which is the addition of PDP models projected by [72]. The clients store many replicas of the data file in the storage which is verified by challenge – response protocol. At the time of the challenge each single unique copy should be produced. The data owner calculates and verifies with the time t the storage of a single replica and w this it verifies for all copies that reside in storage system. All the replica copies have original file are masked randomly by using pseudo –a random function (PRF), all replica is unique in nature and uses different PRF and does not compare with each other. The single sets of tags used to verify by modified homomorphic verification tags of PDP [72], all tags are generated at once for the original data file. It unmasks the existing replica and places it with new one randomly, this very efficient. With all relevant parameters in single replica PDP scheme a multiple replica PDP(MR-PDP) scheme is built very efficiently [74].

The probability of data loss is immensely decreased and availability of data is increased by using MR-PDP. Only the client is authenticated to perform the verification process for integrity and availability hence private verifiability is achieved.

With the multiple replicas along with the remote data possession protocol verification is done publicly to assure the availability, integrity and confidentiality [75]. When the client run out of time, they hire a third party to do the verification behalf of them (TPA) which make it very flexible. The homomorphic authentication along with BLS [65] bilinear signature is employed in this protocol [64]. To support public verification of multiple replicas [76] used a PEMC-PDP protocol which ensures the privacy from third party who verifies the data integrity. To improvisation of the security proposed in [76],a multi copy privacy preservation on serverless storage is proposed by [77].the characteristic of privacy preservation in public auditing makes more flexible and commissioned the possession of data file for verifier where the verifier is unaware of the content in it.

The issues in the multiple replica-based protocols are they need large space which leads to communication overheads. But it achieves the availability and integrity of the data store along the servers.

To overcome the short coming of the replica-based scheme in [75],[77], a Proof of Retrievability (POR) protocols is proposed

The complementary approach of PDP is POR, which allows the verifier to check the data possessed in server by incorporating challenge response protocol. The verifier can reconstruct the whole data with the response given by the server which makes POR more reliable and stronger than PDP. The encoding done on the data file before storing in the remote server using erasure codes. Only static data are concentrated by the POR protocol.

A scheme which is suitable for the internet based operation is proposed by the [78] for clients who store their data in the server. There are n servers where the file F blocks are dispersed using (m, n) dispersal of information [79]. To recover the file, for any m out n fragments are enough to rebuild the whole file. Calculation of MAC (Message authenticated code) for each block after encoding is completed. Using the pre computed MAC s the peer perform spot integrity check on one another on data blocks which spot the data loss but does not ensure whether all the data are secure.

Similarly using erasure correcting coding [80] the integrity of the data file stored across the multiple servers is verified using the algebraic signature and hashing techniques. [81]. The blinding of data and parity by XOR operation with pseudo – random stream

in order makes it collusion –resistant. If the secret parity is leaked then, it easy to regenerate the whole files using both erasure code and pseudo random stream. The verifier requests all the distributed servers to return the signature which already calculated in a specified part of the data file. With returned signature it validates the data accordingly by using keys of algebraic encode and stream cipher encryption. The MAC is constructed where the corruption –detection system in on surveillance [82]. There is communication overhead while handling the file access at the server side as the files are linear blocks which accept the challenge.

A proper definition of POR [83]. which is complementary method to PDP [68]. A challenge –response protocol of this POR scheme provides the remote server confidence that the verifier can retrieve or rebuild the whole file from the response which is transmitted by servers. It employed in two phases set up phase and verification phase. In first phase the client fragments the file F into block of chunks (k). With error correcting code implemented in each chunk and elaborates into n blocks. Before outsourcings the data file the erasure code is used if any detection or modification in any parts of the files. The encryption and permutation is done on outcome result to ensure the dependency of the blocks remain secret and these masquerade blocks are called sentinels which are embedded in the outsource file cipher for error resiliency. The hidden sentinels are hidden inside the data block for adversary detection and their operations in the server. Pseudorandom Permutation (PRP) is used by the client request for the random sentinels to check for the integrity of the data. The probability of corruption reflects the sentinel if a block is corrupted or erased data in the outsourced data. POR cannot be implemented for public database like repositories, libraries. At each challenge the sentinel and its position revealed and cannot be reused which makes the challenges to be limited and fixed.

To overcome the fixed and limited challenges and public verification, [84] proposed a scheme with security which holds proof using two POR schemes which is against arbitrary threats as in this model design by [83]. The private verifiability along with the proof of retrievability scheme is the response of the standard model secured by building pseudo random function (PRFs) is considered as first scheme which had long query. In the second scheme the BLS signatures and random oracle model is used which provide response of POR and public Verifiability. The aggregation of proof into one small authenticator value is completely depended on the homomorphic properties. The client

breaks an erasure code file into several blocks and authenticates each one of them with pseudorandom function (PRF) along with the key which is stored in the server. The user verifies by choosing a random challenge using pseudorandom permutation (PRP) and sends to the servers, upon receiving the challenge the server responds to it. The comparison is done by the user with response and the pre-computed values.

This scheme is same as the above framework but it uses BLS signature for public verifiability authentication. The linear combination is been aggregated by structure of theses signatures. It provides security with computational diffie-hellman assumption over bilinear group in the random oracle model.

In the work of [85] and [86] come up with a framework for the POR protocol which uses integrated forward error –correcting codes.

The client challenges to ensure security by extraction the file through the challenge-response interface. There is simple connection between POR schemes and the idea of hard amplification which comes through the study of complex theory in [86]. In contrary [87] shows different parameter trade-offs when POR are designed and show least interest on POR protocols.

HAIL (a high availability and integrity layer) is proposed by [87] which broaden the RAID basic principle to handle the adversary in serverless server.

HAIL provides efficient security and model improvements than the traditional multi-server application of POR protocols which maintains the file originality and availability throughout the servers. The storage resources can be tested and changes its location if failure occurs, this is done by making use of PORs as pillar support [87],[84].[83]. HAIL proves the client by enabling the set of the servers via challenge –response protocol that holds the file F fully intact and more in particular, that the client can rebuild or recover from any loss with high probability and minimum overheads. HAIL is very sensitive, even if a single bit is prone to security risk which drastic loss over period of time; it protects from the adversaries. The limitation of [87] is not good at public verifiability.

By integrating the robustness based on spot checking in the framework of Provable Data Possession (PDP) is proposed in [66] which focus on robust remote data checking. To guarantee the audits conducted on the server, the integration of forward error –

correcting codes (FECs) with remote checking is done. The encoded file using FEC and PDP is applied on that encoded file F. If the server finds the file is corrupted, ability to detect the flaw is provided by the original PDP framework.

The robustness is achieved by combining the file along with FEC and PDP scheme.

It provides protection against corruption for large and small part of encoded file. If the server is corrupted more than a fraction of encoded file the client will detect with the high probability for the larger portion of encoded. If the server is corrupted at most a fraction of the encoded file, the client will recover the data with high probability for small portion of encoded file. The remote data checking must guarantee the securities as the design are complex and subtle. The vital problem is the choice of the FEC code, how the encoded parameters are selected and the output data layout. The encoded data rate and I/O performance should be maximized on the remote storage which minimizes the overheads faced like I/O complexity of remote data auditing and duplication of data. The choice of encodings is very important which guarantees the security and performance. Some of the FEC codes [83] and [84] are not optimal and experience poor I/O performance.

The idea of mitigation arbitrary amount of data corruption in the robust auditing scheme is proposed in [66], which include both detection of data corruption and to be resistant to it. The data owner replaces the data from the redundant data when they encounter any data corruption or data loss.

An idea is proposed [88] which improves the output of the [66] on robust auditing which incorporate the forward error-correcting codes and remote data checking. The remote data checking along with the encoded file using the FECs to be robust and provide an elaborate analysis of the reliable encoded result which used to measure the probability of the attacks on spot checking.

For the large set of data they use a PDP (E-PDP) which is described in [68]. It shows experimental results along with probabilistic Possession which assure the security by in depth evaluation of auditing which trade-offs the space, performance and security. In the above all protocols the confidentiality of the data is not addressed, only it provides the methods for efficient audit, assurance on the data integrity and availability of the remote data with erasure coded embedded .

For efficient remote data possession in the serverless servers [89] provides confidentiality, integrity and availability of the data. In the term of computation and communication it is very efficient and effective and the verification is done without the challenger to compare the original data and the response sends by the server.

It is the client responsibility to maintain the keys used in encryption to protect the data and it brings key management problem which lead encrypting the keys for security again making the problem worse than solving it in these schemes [64],[67],[83],[89].

To eradicate the key management issues for the client to ensure confidentiality, availability, integrity of the data, the researchers have been done and secret sharing protocols are considered.

For validation of the integrity, recovering the data, safe guarding the confidentiality from unauthorized users and management of keys issue is solved by this protocol.

There is an alternative way for protecting data confidentiality and eliminate the key problem is given by [90]. The secret sharing scheme introduce (m, n) where n is data which are partitioned into n shares and distributed among m nodes. The original data can be reconstructed by m shares i.e. (m<=n), when the data is accessed. The information of the original data is not derived with less than m shares. The system can be highly secure and always available with large n and some reasonable m. The only issue is there are storage flaws where only share same size of the original data.

For the remote data [91] proposed a diminutive secret sharing scheme to provide confidentiality and availability. The traditional secret sharing given by [90], encryption, and dispersal of information (Rabin 1989; Radu 2005).

A short secret sharing like [91] is build by [92] which combination of secret sharing keys and erasure code in the serverless environment to calculate and analysis the performance of the data stores for availability and confidentiality, where the storage resources are from different providers with different geographical location. Based on [93] to achieve the data confidentiality and availability there is scheme designed (SSS) which provides dependability and high performance. For each update server to server communication is avoided by considering multiple – version-based access algorithms. To achieve a secure SVR with the guarantee of wait-free read in such way algorithms are design.

update share a version number is given which is unique which is efficient and satisfies the regular semantics while accessing it. A unique version number (UVN) is developed to access for the client independently to reduce the cost of each update. However, there are incurring potential performance problems when it is implemented distributed serverless storage.

There are two level serverless storage which address the issue by [91], First the server is divided into several groups based on their location information. Using SSS on each data object within each group and the data is shared among the servers in the group. Replication of the shared data to other different groups and the lazy update can be applied to the updated share in one group and forwarded to the other groups. This reduces the response latency for the update access which the user perceived. In Each group independently the consistent share verification is performed. The cost is reduced as server-to-server communication can be inhibited within the group. Thus, the two-level approaches are more scalable than data partition approach.

Using data grid support a secure, robust and high performing storage with dynamic replication by data partitioning. The problem is complex by using the erasure code scheme or secret sharing and/or replicated. The topology is designed in two layers. Multiple clusters form a network topology which illustrated in a general graph. Within each cluster the topology is depicted by tree graph. The share replica allocation has two problems; First is the Optimal Intra need cluster share allocation problem (OISAP) which finds out the cluster shares replicas and it determine its number of shares needed and how it is placed in the cluster. The aim is to develop, a placement algorithm to allocate the shared replicas which significantly reduce the communication cost and access latency.

Public verification is proposed by [94] for dynamic sharing protocol for data storage security. The validation of the shares can be verified by the shareholders where the data shared among the group of servers. Some server store the share which is backup when there is threats and leads to dynamic recovery Additive sharing and verifiable encryption is used by this protocol to complete the recovery and verifiable functions. The main disadvantages of the above Short secret scheme (SSS) are that it does look into the remote data integrity in the servers.

The solution to the problem of information verification of data integrity on distributed storage is proposed by [91] against attacks, storage issue and efficient communication by using cryptographic methods. Using Information dispersal algorithms (IDA) [79], the missing data information are dealt and solve general secures the information dispersal problem and recover from the attacks and loss of data due adversary. There is no use of public or private keys; it does not require secret keys. Recovery of distributed data can do by any party within the system and does not allow to modification or data theft as long as parties are reliable and be honesty. Using finger prints are used along with the cryptographic techniques the solution is found. To maintain the integrity of data they use a "paradoxical" property which holds public fingerprints. With these finger prints anyone can compute using the same unction but can't cheat or forge them.

Hash values are not very appropriate which is used to check the data originality in distributed system data. The hash values are stored by the verifier to check the integrity of distributed data. First, the verifier sends a request to the data storage to get ready for the challenge. The computed hash value should match the value which already stored, if not the data are being modified, this incurs communication overhead due irresistible data communication. Space complexity occurs as large volume of hash values must be stored by the verifier. If there is failure at a single point, the checker is collapse and this method is not reliable. So to overcome this light weight process of integrity check with public verification is mandatory.

Dependability and integrity of the data to ensure confidentiality with lightweight process is proposed in [95] which more flexible and efficient for dynamic data checking the integrity scheme. The scheme is publically verifiable with consistent data shares in the distributed storage system. The data –originating senor partitions of the original data into several multiple shares are done by use erasure coding [96],[97] and secret sharing techniques. Compared to the traditional replication proposed in [67],[77] the construction significantly reduce the communication and storage flaw and achieve dependable data storage by duplicating the data as in original data sets. By utilizing algebraic signatures which are encouraging algebraic properties and spot checking for ensure the data integrity and availability [81],[98]. This allows the shareholders to verify data integrity on dynamic data in a randomly with minimal problems. In each check the data–originating sensor appends a discrete parity block to all data share which makes the shareholders to perform the verification on the distributed data share

independently. The important feature of this scheme is that it provides zero false negative probability. The unauthorized party can be detected with the high probability in every single operation. Verification of aggregated data share for its integrity with high probability is explained [99]. For secure and dependable storage for distributed system scheme for UWSNs is shown in [100]). The advantage of secret sharing and reed Solomon code are low communication overheads and computation security with shorten data dispersing size. A linear coding method is incorporated instead of Hashing function to public verification in distributed manner which gives very low communication and storage overheads without a holding the original data. Hence the data confidentiality, availability and integrity are achieved by using the protocols with public verification but it is employed for the static data in the storage system [99],[100].

The operation like update, delete, append and insertion done the data stores in serverless server are dynamic data which must verified in a period of time to make more secure and reliable. For dynamic data verification there, highly efficient design has been proposed [72]. Provable secure PDP with data dynamic support are done by "Scalable Data Possession" which is based on symmetric key cryptography technique which eliminate the number of encryption . There are token computed before outsourcing the data, where each token is covered by random set of data block and the original data is stored in the server along with it. To obtain the proof of the data possessed by the server, it challenges the server along with random block indices. Once the server receives the challenge it computes a small integrity check over the mentioned blocks and sends the response to the client. If the response matches the corresponding value pre-computed by the client then the data is secure or not. In this scheme the client either keep the pre-computed token locally or outsource in encrypted form to the server. It improves the PDP in the terms of cost, computation flaws and bandwidth considerably. There are disadvantage in this approach where the client can perform limited updates and challenges and it fixed priori. Insertion is difficult and cannot insert a data as desired, furthermore for each update all the process of computing the token is repeated as new entry which make it more complex for bigger files. Since it based on symmetric key cryptography, it does not support Public verifiability.

Effective and flexible storage verification for distributed system is proposed by [93],[99]. This scheme works explicitly on dynamic data to ensure the authenticity and availability of data in the serverless server. The system design is against the byzantine

failures, redundancies of the data during the file distribution preparation using erasure code build using reed Solomon code. Comparing with traditional replication file storage in the distributed system techniques the construction reduces the communication and storage space overheads. Homomorphic tokens are utilized with distributed file along with the erasure code data which ensure storage correctness insurance and fast error localization. When the errors are detected, the scheme ensures immediate action on verifying correction and locating the misbehaving server. Exploration of algebraic properties of the token computing and erasure code is needed to maintain the balance between error abolition and data dynamic operation. It shows how efficiently it support dynamic operation on the data blocks and also preserve the correctness assurance. However, like mention in [69] the insertions on the index position s of the data block are not supported.

To support data dynamics based on PDP model [68], construction of two efficient PDP by [101]. It supports the dynamic data operations on the data storage at block level and by using rank based verification skip list which not, the index position in the serverless server using RSA trees [102],[88]. Rank based authentication with skip list is mainly used to authenticate the information in the tag of the blocks which are updated or undergone challenge [68]. In the DPDP scheme said by [101], the files are being fragmented into m blocks and tag is computed for each one of them. The representation of the block is stored in j^{th} bottom –level node of the skip list.

The tag attached is stored data protects the integrity of the file block. In the challenge phase the client requests the server prove the correctness by selected random blocks. Upon receiving the request, the sever send s a tag along with search paths and send the combined blocks as part of the challenge issued. The client verifies the path along search if the block tags using the information of the metadata which attached at the start node. The metadata is compared with the server response where the client acceptance based on the search path which is verified. The skip list which is authenticated are utilized to modify, insert and delete a block tag by achieving the dynamic operation of the data file. The efficiency of the scheme is remaining doubted and unclear.

A BLS signatures based on homomorphic authenticator is proposed by [103], which enables full data dynamics using Merkle Hash tree (MHT) explain in [104] in its place of skip list given by [101] for verification of integrity in serverless servers. The client

uses encoded file done by incorporating Reed Solomon codes [80], which is divides the files into n blocks. Then using BLS signature the client compute signatures and generate the public and private keys. The construction of MHT done by the client who assigns the root is R, and leave nodes of the tree are file tags. The private key is assigned as the R root by the client. The client send the file along with the root, signature to server then deletes the local copy of it.

The integrity check can be either done by the client or the third party auditor (TPA) where they are outsourced to challenge the server randomly. The TPA first uses the private key to check the signature of the root before challenging the server. It state "FALSE" if the verification fails and rejects it or otherwise it recover the public key. By mentioning the position of the blocks the client can check the integrity of the data using a challenge message chal either generates by TPA or the client. The server receives the challenge from TPA generates response along with some amount of auxiliary information such as the path from root to leaves and sibling nodes in MHT. The comparison is done between the new root which is generated with response given by the server with the old one. It the results are different, and then the verifier shoes "FALSE" or if the proof is verified using BLS signature shows "TRUE". Thus the data dynamics is supports the operations like block modification, add, delete. For update operation, a request is sent by client to the server and update operation is performed as per the request and alter the MHT. The disadvantage is that replay attack which makes the data block audit system insecure while updating with the same hash value.

A work is proposed by [89], to eradicate the replay attack for the remote data. The integrity checking for remote data with dynamic support using index table is done. Using random sampling, index hashing and fragment structure they construct a effective audit scheme for dynamic data verification which outsourced. A secret key and public key of the system is generates by the client who owns the data, the file is pre- processed by the secret key which consist of n blocks ,a set of public verification parameter(PVP) and a index hash table which are stored in third party. The TPA transmits the file along with the tags to the serverless service provider and deletes the local copy. A "random Sampling "challenge is issued by the interactive proof protocol of retrievability in order to audit for any security breach. The index –hash table (IHT) can be updated and manipulated by authorized application of the data owner with the secret key which is stored in TPA. The checking algorithm ensures the secure storage

from unauthorized threats by using private secret key which identifies the forge records. A cooperative Provable data Possession for Integrity check on multiple cloud storage is given by [68]. The protocols proposed by the above is not suitable for the real serverless computing solution which require trustworthy servers. The assurance of the confidentiality of data is lacking in the above protocols discussed though it supports dynamic data auditing for integrity check and always available on remote distributed data servers.

Privacy –preserving a public auditing scheme for cloud storage is proposed by [41],[103]. The server has choice to select a random number embed along the data block and send to the servers instead of sending the linear combination of data block to the auditor directly. Masking of random number is done by the server and sends back to the client verifier with the linear combination of data block and random number in order preserve privacy of the client. The privacy is maintained from the outsourced TPA and it does not assure the confidentiality from adversary inside as the data are not encrypted before it is delegated.

The data are being protected from malicious attacker from internal and external by this proposed scheme by [76]. It uses dynamic multiple data copies maintained in the serverless servers based on MR- PDP. There are two schemes which ensure integrity, availability and confidentiality they are tree based and map based dynamic multi-copy provable data possession. The tree based scheme is the fine-tuned extension based on reliable data structure to provable possession of dynamic single copy. Generation of multiple copies of the data file by the client is done then each copy is encrypted and fragmented into sectors where each one of them belongs to one prime. For multiple copies of block, the tags are generated separately, and then MHT is also generated by the owner for copy of the file. The hash values are the leave nodes of MHT and the directory is generated by using the path from root to leave node. The dynamic operations are supported by using map version tables instead of trees. The serverless service providers prefer MHT to maintain the server to reduce the overheads. For the mobile devices in cloud computing, a provable data possession of resources – constrained mobile device in cloud computing is proposed by [105]. These mobile terminals only needs secret keys and random numbers generated with the help of the trusted platform model chip. The mobile devices must be capable of handle the work load while computing and space needed for it. Similar scheme used in [103] by integrate

the bilinear signature and Merkle hash tree (MHT) which comprehensive the token verification of the all the file into small signature to limit the communication and storage issues.

For different systems the probabilistic verification schemes attain security if it achieves the following parameter like availability, integrity and confidentiality. There are some flaws which ceases the system to achieve the requirements which affects the efficiency of the data file stored in serverless storage servers. Using pseudorandom sequence, the integrity of the data is verified by using probabilistic protocols, which does not cover the entire files works on probability that leads to data files missed for challenges. Since the integrity proof is not got for all the files, data may be corrupted and slip away without detection. For this the server have to perform many numbers of challenges to cover the entire files to obtain high probability which not feasible. The spot checking is not practical to achieve lightweight process and integrity assurance is not satisfactory.

CHAPTER -III

IDENTITY BASED CRYPTOSYSTEM FOR SECURED AUTHENTICATION

CHAPTER 3

IDENTITY BASED CRYPTOSYSTEM FOR SECURED AUTHENTICATION

3.1. INTRODUCTION

Serverless computing is a cloud model that enables the developers to manage the servers and decide low levels of infrastructure [106]. It offers a real, pay-by-the-go, resource-free service and it reduces the limitations for developers by providing strong trust on cloud providers with operational complexities [107]. Compared to the other cloud models, serverless model is closer to the cloud computing expectations that should be treated as a utility's services [108]. It is a becoming of compelling paradigm for cloud applications, mainly because of the architectures have recently shifted to micro-services and containers [109].

Serverless computing is a paradigm that represents a novel evolution on cloud models, its platforms and abstractions [106]. The serverless computing uses explicitly the function as its core deployment unit, However, the functions from different users running on a shared platform is considered critical during isolation [48].

Hackers or the illegal use of sensitive data or resources are always increasing security concerns with sensitive user data [110]. Encrypting whole authentication is popular for the improvement of security problems. For such encryption data search, index terms are used. In this case, the retrieval entity is unable to read the data.

Several encryption systems exist, some of the models are public key encryption system [111]; Biometric based Authentication [112]; attribute based encryptions [113]; Quantum Authentication [114], asymmetric image encryption system [115], State estimation-based dynamic encryption system [116], memory encryption and authentication secure against side-channel attacks (MEAS) system [117], Optical encryption and sparsity constraint authentication [118], attribute based encryptions on mobile ad-hoc [119], authentication with blockchain and text encryption protocol [120], Image authentication using double image encryption [121], encryption technique on

cloud computing [122] , where complex encryption operations are mainly involved on event of any anonymous user attacking the functions from different users are exploit the advantage of identity based cryptosystem in the Function-as-a-Service (FaaS) serverless model to improve the privacy of data.

3.2. PROPOSED IDENTY BASED CRYPTOSYSTEM MODEL

In this section, an Elliptical Curve Cryptography (ECC) based identity of the user and the function being deployed in FaaS model. Initially will check the feasibility of ECC in improving the data security based on the user identity. If the ECC algorithm fits, the algorithm will be improved in a better way such that it suits the functions of FaaS and does not affect the performance of FaaS.

The architecture of the serverless model with FaaS model is given in Figure 3.1.

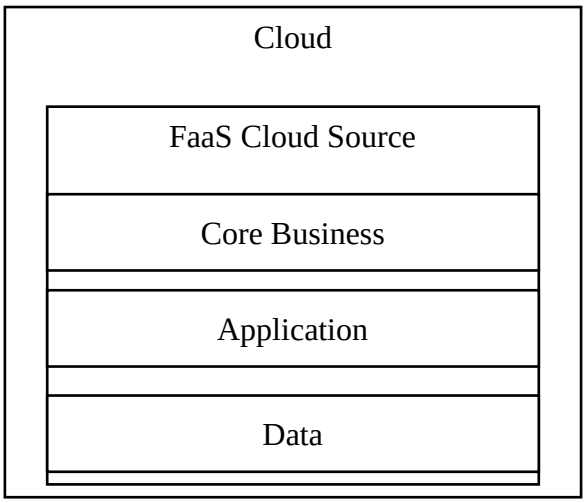


Figure 3.1: FaaS with serverless architecture

3.2.1. FaaS MODEL

FaaS is a serverless computing implementation through serverless architecture. The serverless architecture is dominated and was widely considered to be a serverless computing. As a result, we will not distinguish between them and use these two terms interchangeably.

FaaS release developers from heavy implications by executing code as a response to an event, so that modular chunks of functionality can be simply uploaded to the cloud that can be run independently for the deployment of the service.

3.2.2. STRUCTURE OF AUTHENTICATION MODEL

The proposed model has four phases, as follow as,

- **Setup phase**: The universal feature set is defined in the setup phase. In this phase, the authority takes the input and output of the system's public parameters and system master keys by implicitly securing the parameter.
- **Encryption phase**: This phase is carried out by the sender and includes input from public parameters, message and access structure. Encrypts the message under a specific access structure, so that the message can only be decrypted by recipients whose attribute sets fulfil this access structure.
- **Key Generation phase**: The authority runs this phase, which takes system master keys and the user's attribute set as the input and releases the corresponding secret key.
- **Decryption phase**: This phase is performed by the data recipient that includes the ciphertext, secret key that matches its own set of attributes and public parameters. If the attribute set of the receiver meets the ciphertext access structure, then the message is successfully decrypted in this phase.

3.2.3. ECC MODEL FOR SECURITY IN FaaS

The only known approach for ECC breaking is to solve the ECDLP problem. The best known solution is an enhancement, where *n* is the elliptical curve in question, involving estimated measures. By, this measure and anticipated future processor speed, the size of the key requested for different public-key asymmetric systems has been calculated to match the security levels of a symmetrical private key systems.

Table 3.1: Primitive Notation

Notations	Description
U_i	Particular User
A	Adversary
$h_1(\cdot)$, $h_2(\cdot)$, $h_3(\cdot)$	Three one-way hash functions
ID_i	Identity of U_i
s, $PK = s \cdot P$	Private and Public key pair of T
T	Trusted Third Party

Even if the key size is reduced as conventional technology, the ECC algorithm provides the same level of safety. The mathematical operations in ECC are derived from the equation

$$E_p(a, b) : y^2 = x^3 + ax + b \bmod p \qquad (3.1)$$

Here a and b are used to define the elliptical curve, and x, y points, even if the point at the end lies on the curve, when it complies with the previous phase.

INITIALIZATION PHASE

The trusteed third party T is used for assembling ECC parameters during this phase. Initially, an elliptic curve is considered to be $E_p(a, b)$ and the random base point (P) is collected by T with the help of $h_1(\cdot)$, $h_2(\cdot)$, $h_3(\cdot)$. Thus, by generating the secret key(s), parameters $\{E_p(a, b)P, h_1(\cdot), h_2(\cdot), h_3(\cdot)\}$ are revealed in T.

AUTHENTICATION PHASE

Each user must be authenticated with other connecting nodes in order to establish communication with other nodes. As the following procedure is shown: authentication between the users

Step 1: $(U_i \text{ and } U_j) = U_i \rightarrow U_j : \{X_i, Y_i, K_{ip}, ID_i, t_i\}$

The user U_i is connected with K_{ip} and K_{is} and that tends to collects the information $x_i \in Z_{p.}$ The time stamp t_i is then set up and the values X_i and Y_i is found using the user U_i. This tends to places a request for authentication to the user U_j with proper transmission of information K_{ip}, ID_i, X_i, Y_i and t_i.

$$\boldsymbol{X_i = x_i \cdot P} \qquad \textbf{(3.2)}$$

$$\boldsymbol{Y_i = x_i + K_{is}\, H_2(ID_j, X_i, t_i)} \qquad \textbf{(3.3)}$$

***Step* 2:** $\boldsymbol{U_j \rightarrow U_i : \{X_j, Y_j, K_{jp}, ID_j, t_j\}}$

When an authentication request message is sent from U_i to U_j, U_j checks the time stamp. The session is terminated if the difference in time stamp between U_j and U_i exceeds the threshold level. If the threshold level is less than the time stamp difference, U_j will accept the authentication application from U_i. If the application is confirmed by Yi As soon as possible, Uj will move to the next process, or the session will be completed. For the shared session key, user U_j calculates X_j and Y_j, where $SK_{ij} = H_3(x_j \cdot X_i)$. The message is thus transmitted using the user U_j with the security parameters K_{jp}, ID_j, X_j, Y_j, t_j.

$$\boldsymbol{Y_i \cdot P = (K_{ip} + X_i + H_1(ID_i, K_{ip})s \cdot P) \cdot (H_2(ID_j, X_i, t_i))}, \qquad \textbf{(3.4)}$$

$$\boldsymbol{X_j = x_j \cdot P} \qquad \textbf{(3.5)}$$

$$\boldsymbol{Y_j = x_j + K_{js} H_2(ID_i, X_j, t_j)} \qquad \textbf{(3.6)}$$

STEPS OF AUTHENTICATION

The establishment of authentication between user A & B is given by:

Step 1: Let d_A is the private key of user A and d_B is the private key of user B. The private keys are established using a random number, which is lesser than n i.e. domain parameter.

Step 2: Consider $Q_A = d_A G$ and $Q_B = d_B G$ are the two public key of A and B respectively with G as the domain parameter that estimates the private keys.

Step 3: Exchange the public keys between A and B

Step 4: The entity *A* estimates $K = (x_K, y_K) = d_A Q_B$

Step 5: The entity *B* finds $L = (x_L, y_L) = d_B Q_A$

Step 6: If $K=L$, then the shared secret key is selected as x_K and then the transactions are confirmed.

3.3. RESULT ANALYSIS

The simulations are carried out in terms of different performance metrics with other existing methods. The experiments are conducted on a desktop computer with 3.4 GHz processor on an Intel core i7 with a storage capacity of 500GB on an 8GB RAM. The experiments are conducted with different file sizes that ranging between 750-10000 kb. The computations are carried out for several iterations and the average values are determined.

Table 3.2 and Figure 3.2, shows the results between the total number of rounds and execution time. The results of simulation shows that the improved ECC based serverless security has reduced execution time than ECC and ECDSA algorithms with respect to increasing number of rounds in the system.

Table 3.2: Execution time (ms) based on number of nodes

Rounds	ECC	ECDSA	Identity based Cryptosystem
100	151	125	113
200	195	169	158
300	239	213	202
400	284	251	239
500	328	289	277
600	372	326	315

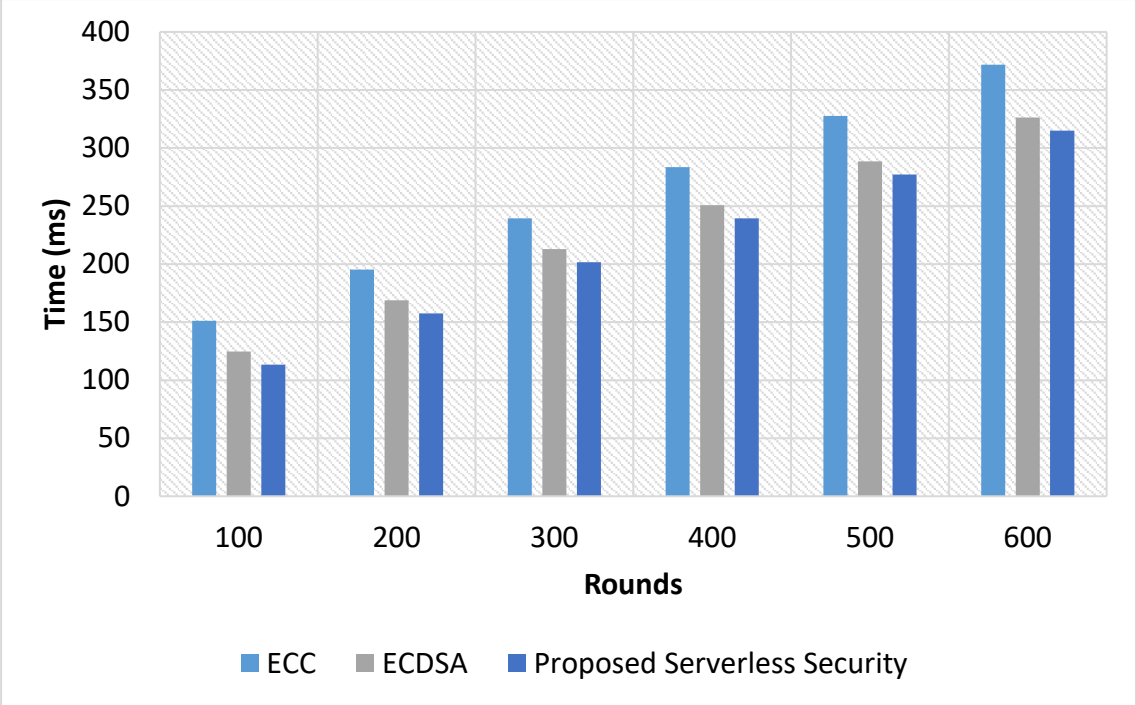


Figure 3.2: Execution time (ms) based on number of nodes

Table 3.3 and Figure 3.3 shows the results between the file size and execution time. The results of simulation shows that the improved ECC based serverless security has reduced execution time than ECC and ECDSA algorithms with respect to the increasing file size.

Table 3.3: Execution time (ms) based on file size

File size	ECC	ECDSA	Identity based Cryptosystem
750	154	63	53
2000	349	252	236
4000	490	340	307
6000	622	504	460
8000	760	630	583
10000	1008	781	747

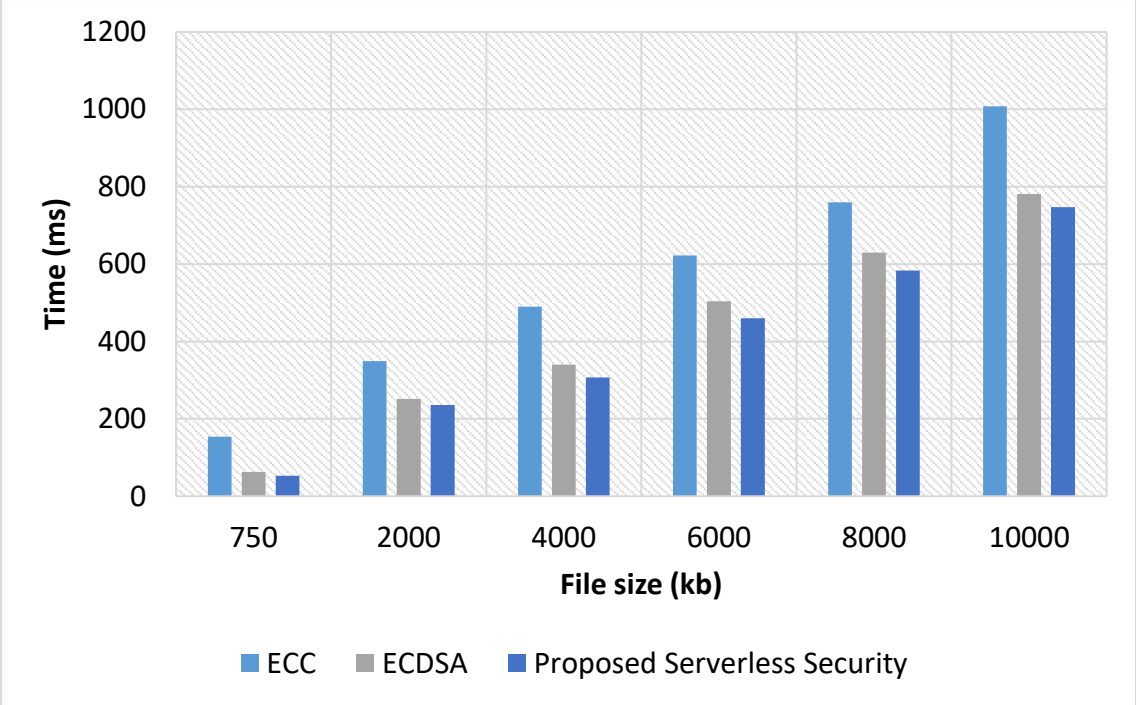


Figure 3.3: Execution time (ms) based on file size

Table 3.4 and Figure 3.4 shows the results between the file size and encryption time. The results of simulation shows that the improved ECC based serverless security has reduced encryption time than ECC and ECDSA algorithms with respect to the increasing file size.

Table 3.4: Encryption (ms) based on file size

File size	ECC	ECDSA	Identity based Cryptosystem
750	126	55	29
2000	315	239	239
4000	406	343	315
6000	577	451	418
8000	741	566	548
10000	861	796	760

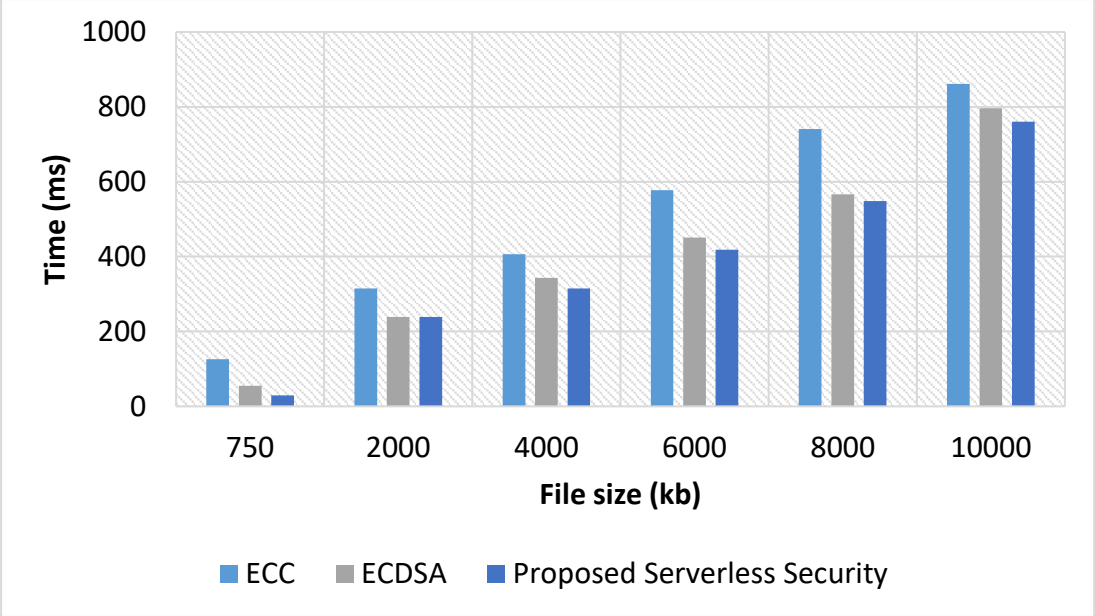


Figure 3.4: Encryption (ms) based on file size

Similarly, the Table 3.5 and Figure 3.5 shows the results between the file size and decryption time. The results of simulation shows that the improved ECC based serverless security has reduced decryption time than ECC and ECDSA algorithms with respect to the increasing file size. It is to be noted that there is a marginal difference between the encryption and decryption time during encrypting and decrypting the authenticated information.

Table 3.5: Decryption (ms) based on file size

File size	ECC	ECDSA	Identity based Cryptosystem
750	127.2	56.1	30.1
2000	316.3	240.5	240.5
4000	407.5	344.8	316.7
6000	578.8	452.3	419.3
8000	742.3	567.6	549.6
10000	862.7	797.7	761.7

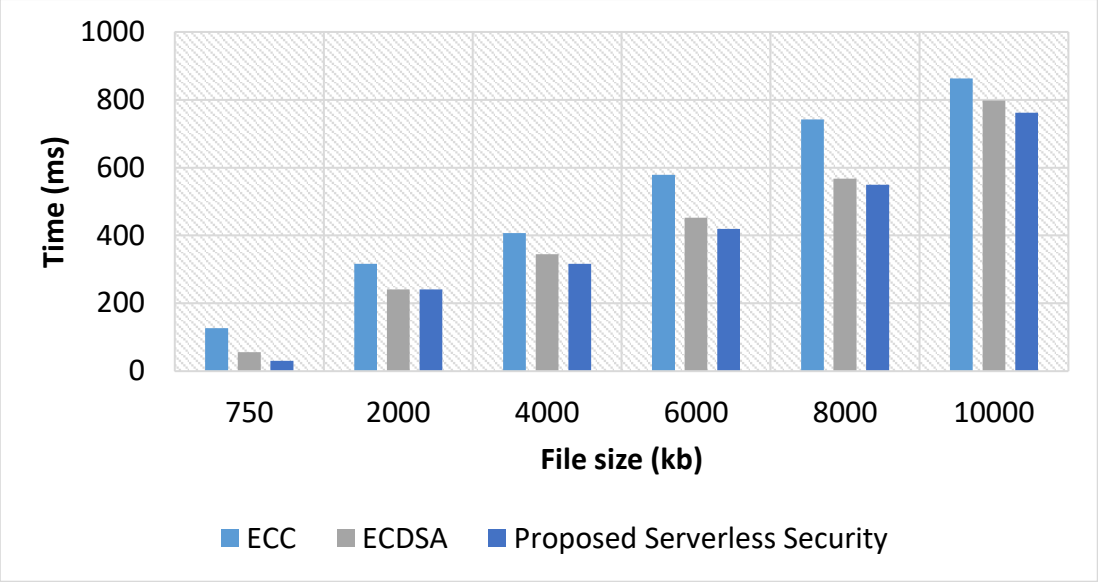


Figure 3.5: Decryption (ms) based on file size

Table 3.6 and Figure 3.6 shows the results between the file size and throughput. The results of simulation shows that the improved ECC based serverless security has increased throughput than ECC and ECDSA algorithms with respect to the increasing file size.

Table 3.6: Throughput (kbps)

File size	ECC	ECDSA	Identity based Cryptosystem
750	0.1260	0.8820	1.3860
2000	0.9320	1.5750	1.8900
4000	1.6670	2.1420	2.3940
6000	2.1960	2.5200	2.7720
8000	2.8210	2.8980	3.0530
10000	3.1410	3.1500	3.4600

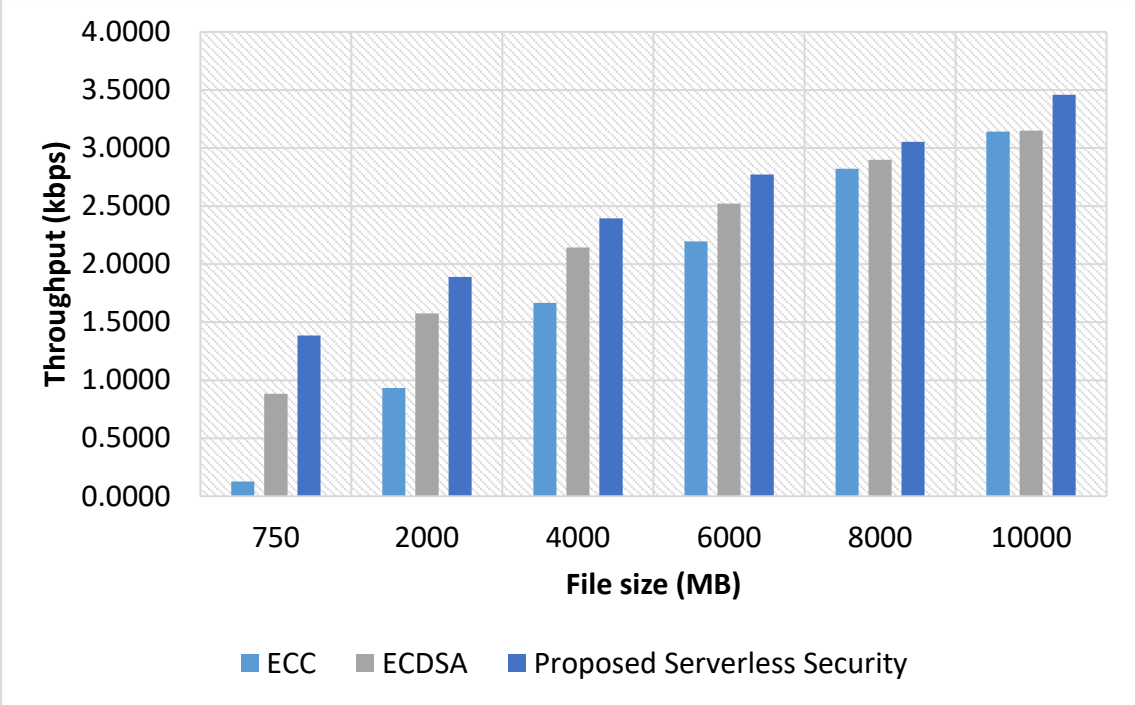


Figure 3.6: Throughput (kbps)

Table 3.7 and Figure 3.7 shows the results between the file size and power consumption. The results of simulation shows that the improved ECC based serverless security has reduced power consumption than ECC and ECDSA algorithms with respect to the increasing file size. The results of entire simulation shows that the proposed method obtains improved security of data in FaaS serverless model than the other methods.

Table 3.7: Power Consumption (mJ)

File size	ECC	ECDSA	Identity based Cryptosystem
750	143	84	35
2000	199	140	103
4000	275	200	173
6000	328	247	243
8000	414	322	294
10000	437	370	356

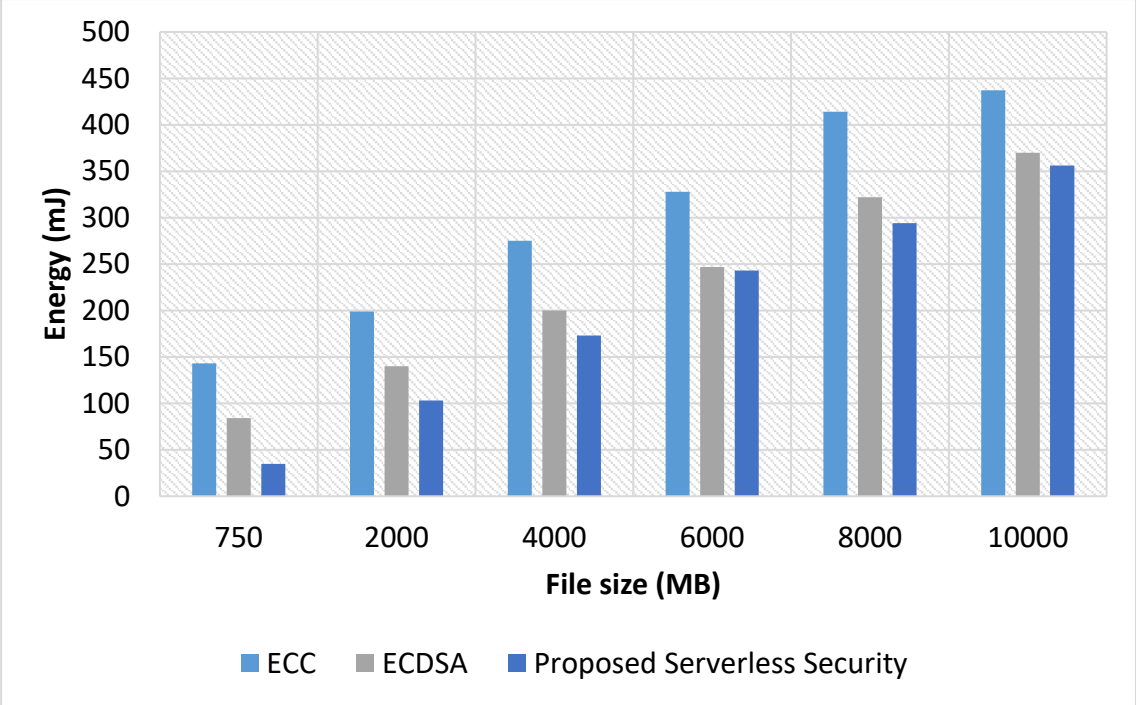


Figure 3.7: Power Consumption (mJ)

CONCLUSION:

In this chapter, ECC based user authentication is proposed in terms of user identity and FaaS model. The system avoids the anonymous users attacking the functions of FaaS from different users. The ECC based cryptosystems exploit the identity based cryptosystem in the FaaS serverless model, which improves the authentication thereby increasing the data privacy. The simulation results confirms the improved performance of the model than to the conventional models. The simulation results show that the proposed method obtains improved security of data in FaaS serverless model than the other methods.

CHAPTER -IV

ATTRIBUTE BASED ENCRYPTION IN SERVERLESS COMPUTING

CHAPTER 4

ATTRIBUTE BASED ENCRYPTION IN SERVERLESS COMPUTING

4.1. INTRODUCTION

Serverless computing is considered as a partial realization of an event-driven solutions, where the applications are defined based on its events and actions. Serverless computing is also a reminiscent of a database and general computing systems, where the actions are reactively processed in event streams [123]. The function of platforms in serverless computing embraces fully the ideas, which helps in proper outlining of action using function abstractions and thereby establishing event processing logic [124].

Serverless computing proves to be a better option for IoT applications, where it gets intersects with the fog or edge infrastructure. There exist several efforts to integrate the serverless computing into a datacenter hierarchy, to facilitate the anticipated IoT devices growth [125]. The AWS on this field allows the application developers to fix limited Lambda functions in edge nodes [126]. Further, AWS pursued serverless computing expansions [127] that improves the functionality of programming in single IoT model. Application developers helps in decomposing a large applications into small functions without the help of server computing that makes it possible for applications components to scale individually, but this poses a new problem in consistent management of a wide range of functions. Step functions have recently been introduced by AWS [128], making it easier to organize and view the function interacts.

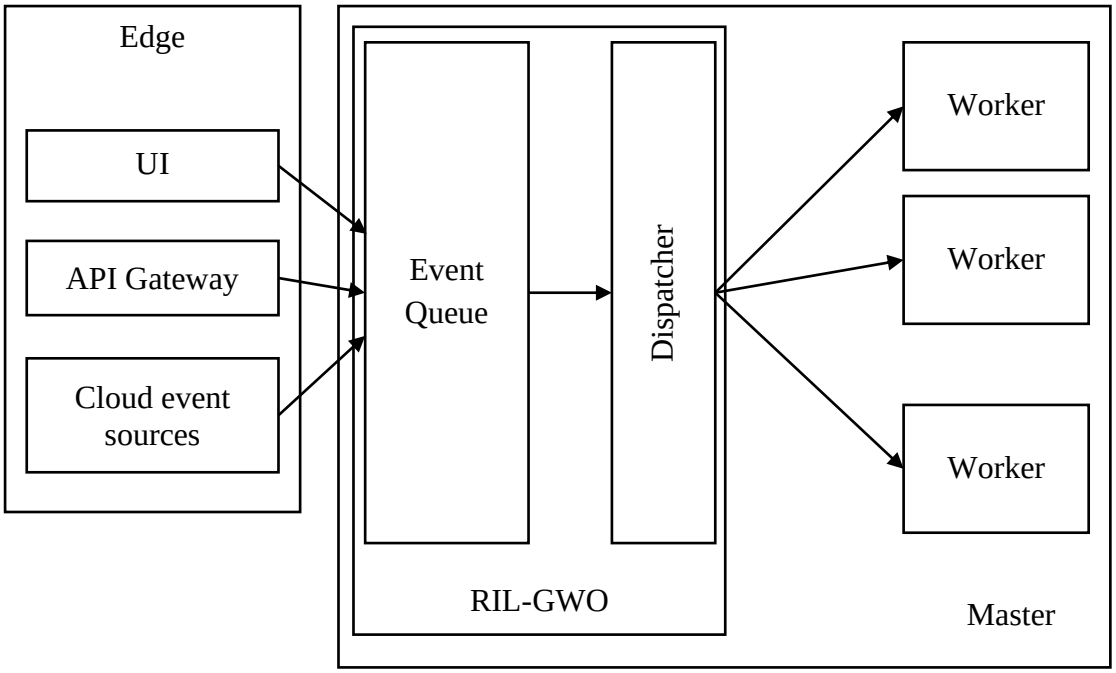


Figure 4.1: Architecture of Serverless Computing

Strong function isolation is essential as many users' functions operates on a shared platform. In addition, serverless system security is also an open investigation issues like Optimization [129], Reliability [130], Security [131][133], Storage [132], security over the Network [134]. It is a harmful proposition to host arbitrary user code in multitenant systems in containers, and attention should be taken to prevent vulnerabilities when constructing and operating function containers.

The security containers and Remote Procedure Calls (RPC) intersection is a real testing model for container security. While serverless platforms are capable of fitting the function containers and arbitrarily restricting the permissions of function, and the further study is used for evaluating the attack within the environment with functional execution [135].

As hackers exist or the sensitive data or resource is used illegally, the security problems on sensitive user data are increasing all the time. There exist four different safety requirements that should be addressed before any data is stored in serverless computing.

i. The serverless dispatcher as in Figure 4.1 should, first provide the user sensitive data with confidentiality of the information.

ii. Secondly, confidential data are at risk, as the user is permitted to access the data only for data access. The user identities must therefore also be carefully handled.

iii. Thirdly, there exist multiple intruders and malicious users, where the dispatcher provides an optimal resistance to ensure the security of information.

iv. Fourthly, before the time of expiration, data is not allowed to access, where it is removed. As hackers can request data access at any time, where the authorized users use an alternative to provide proper access to the data within specific intervals of time and the data is removed from the cloud server after a predefined time. Thus, an unauthorized access to the sensitive information of the user can be less possible.

Encryption is considered to be popular to improve the security issues. Index terms are used for the data search process in searchable encryption. In this case, the entity carrying out the retrieval operation cannot read the responses. There exist several encryption systems available [65],[113], where it suffers mostly from complex cryptographic operations [122].

4.2. PROPOSED ATTRIBUTE BASED ENCRYPTION MODEL

In this section, an access control system using attributed based encryption on serverless computing model is proposed. Initially, the data is encrypted using user attributes, further the data is split into cipher text. It is finally decrypted using a decryption algorithm and then the cipher text is distributed in the network and the encapsulated texts is stored in the serverless system. The data is shared in serverless computing model using searchable encryption process. The data is not allowed to get accessed before the predefined time and it gets expired before the time of expiration. Hence, the unauthorized users are not allowed to access the content in serverless model.

The proposed system uses six different entities for accessing the data in secured manner that includes service provider, owner, user, attacker, third party and server. The service provider offers storage and retrieval of data to its users.

The data is stored in serverless system in an encrypted form, where the encrypted data is then decrypted and the authorized users are allowed to access the original data with the decryption keys and necessary attributes. Attackers are the entities, who tries to access the server without being authorized. They attempt to access the data both before and after the release or expiration time, respectively.

The serverless model protects the data from malicious users, where the third party is allowed to maintain the attributes. The third party is used to generate the parameters of the system including public parameters, secret and decryption key. The time server without interactions records the reference time. It records a precise release time when the key updates, which are time limited. Key shares are managed and it is then stored using the network nodes. The attackers finally attempts to get the protected key shares.

4.2.1. SECURITY REQUIREMENTS

The security requirements used in the proposed study is elaborated.

***Data Confidentiality*:**

Sensitive data is protected from malicious user, who has no access to such data. Access rights involve adequate credentials, where an authorized user satisfies before accessing the data.

***Collision Resistance*:**

Resistance to collisions means that many users are unable to decrypt the encrypted data, where they collaborate and combine the individual key for encryption.

***Attack Resistance*:**

Many attacks are protected against the proposed scheme like Cyber or brute-force. A malicious user in brute-force uses possible key and decrypt the encrypted data and it uses several identities in Cyber attack and then decrypts the encrypted data.

***Non-accessibility of sensitive data before release time*:**

During its authorization period, the sensitive data are accessible before the required time to release and user are not allowed to access the data.

***Delete data after expiration* time:**

The sensitive data are auto-destructed after the time of expiration.

4.2.2. SECURITY ASSUMPTION - BILINEAR MAP

Let G_0 and G_1 is considered as the multiplicative cyclic bilinear groups of prime order p. Let g be a generator of G_0. A bilinear map is a map $e : G_0 \times G_0 \rightarrow G_1$ with following properties:

1. Bilinearity: for all $g \in G_0$ and $a,b \in Z_p$, we have $e(g^a, g^b) = e(g,g)^{ab}$.
2. Non-degeneracy: $e(g,g) \neq 1$.
3. Computability: There is an efficient algorithm to compute $e(u,v)$ for $u,v \in G_0$.

ConsiderG_1, G_2 and G_T forms a bilinear groups with g and h being the generator of G_1 and G_2.

Define $g_i = g^{\alpha i}$ for a unknown $\alpha \in Z_p^*$ and set y = $(g_1, g_2, \ldots, g_n)$ and the algorithm B tends to solve the BDHE problem if the input is g,h,y with an advantage ε.

$$\mathbf{[Pr[}B(e(g_{n+1},h))\mathbf{=1]-Pr[}B(Z)\mathbf{=1]]} \geq \varepsilon \qquad \mathbf{(4.1)}$$

Where

Z is regarded as the random element of G_T and the decision of BDHE assumption holds if the value of ε is considered negligible for a polynomial algorithm.

4.3. PROPOSED ENCRYPTION TECHNIQUE

In this section, a secure model is designed using an access control model in serverless computing environment. This method uses an attribute based encryption to provide data security in serverless computing model.

***System Setup*:**

The input is a security parameter (κ), which has three groups G, G_1 and G_2 with prime order Q, generator g of G, and generator g_1 of G_1. The security technique uses two exponents α and $\beta \in Z_Q$ in random fashion with three hash functions H:

$$H:\{0,1\}^* \rightarrow G,$$

$$H_1:\{0,1\}^* \rightarrow G_1 \text{ and}$$

$$H_2:G_2 \rightarrow \{0,1\}^n;$$

There are three outputs that includes public P_K, master M_K keys and its respective system parameters. Here, the public key is made to be accessed by the users in the network and the master key is made secret only to specific entities, where the notations are represented below:

$$M_K=(\beta,g^{\alpha},x\in Z_Q) \quad (4.2)$$

$$P_K= (G,g,(g_1, y= xg) \in G_1,h=g^{\beta}; \omega=e(g,g^{\alpha})) \quad (4.3)$$

Where,

t is defined as the threshold value,

n is defined as the key shares,

b is defined as the total present and

b_t is defined as the total extraction times.

Finally the system parameters is given as follows:

$$\text{Param} = (\kappa,t,b,b_t,n,G,G_1,G_2,H,H_1,H_2,Q,g,g_1,e,y,\alpha,\beta) \quad (4.4)$$

***Private Key Generation*:**

The key generation ($KeyGen_i$) outputs the key S_K from a user u_i using a trusted authority with an identity ID_i. The Z_Q randomly chooses the random variable r and then the $r_j \in Z_Q$ choses the attribute $\lambda_j \in \Lambda_i$. The private key is generated as:

$$S_K = \left(D = g^{\frac{\alpha+r}{\beta}}, \forall \lambda_i \in \Lambda_i : \left\{ D_j = g^r \cdot H\left(\lambda_j\right)^{r_j}, D_j^{'} = g^{r_j}, D_j^{''} = H\left(\lambda_j\right)^{\beta} \right\} \right) \quad (4.5)$$

***Anonymous Key Generation*:**

The trusted authority (TA) generates the anonymouskey$A_o = H(ID_o)^{\beta}$ for the serverless model.

Pseudonym Generation:

The data pseudonym is generated by choosing $t_1$in random manner using Z_Q and it is represented as $P_o=H(ID_o)^{t1}$, where session key agrees A_o and $P_o$with the user for attribute scrambling.

Data Encryption:

The data *M*is encrypted by proper execution of encrypt algorithm with access tree (*T*) and outputs C_T''. The polynomial q_x is thus selected for each access tree node and then degree (d_x) is set for q_x lesser than threshold value of the node.

Hence $d_x = k_x$-1 and then random value s is selected for a root node *R* from Z_Q and then $q_R(0)=s$ is set. *dR* is selected randomly using the polynomial points *qR* and then $q_x(0) = q_{parent}(x)(index(x))$ is set for other tree node and chooses d_x for other tree points q_x.

The cipher text is thus generated with *Y*as leaf nodes set in the tree *T*:

$$\boldsymbol{C_T'' = (T, C_e = M\omega^s, C=h_s, C''=P_o, \forall y \in Y:\{C_y=g^{qy(0)}, C'_y=H(attr_y)^{qy(0)}\})} \qquad \textbf{(4.6)}$$

***Attribute Scrambling*:**

Attribute scrambling obfuscates the attributes, where the ciphertext exposes the attribute set. The attribute scrambling conceals the entire attribute set in a tree T and gets access to another tree T'. With the attribute set in the tree$S = \{\lambda_i, \ldots, \lambda_k\}$, the attribute scrambling tree is computed as,

$$\begin{aligned} SIT_{o,s} &= \forall \lambda_j \in S : e\left(H\left(\lambda_j\right), A_0^{t_1}\right) \\ &= \forall \lambda_j \in S : e\left(H\left(ID_o\right)^{\beta t_1}, H\left(\lambda_j\right)\right) \\ &= \lambda_j \in S : e\left(H\left(ID_o\right), H\left(\lambda_j\right)\right)^{\beta t_1} \end{aligned}$$

The scm_{attrx} at the leaf node is generated by the $SIT_{o,S}$ instead of λ_x With respect to$attr_x$. Replacement of *T* with *T*′ in C_T'', the new C_T' is thus generated as:

$$\boldsymbol{C_T' = (T', C_e, C, C'', \forall y \in Y:(C_y, C_y'))}. \qquad \textbf{(4.7)}$$

***Data re-encryption*:**

Data re-encryption is used to improve the security that considers *CT*′, K and params as its inputs, and the ciphertext C_T is the final output.

***Self-destruction generation*:**

The SDO algorithm encapsulates the data and then it gets self-destructed using SDG = ($L, C_D, param$), and it is stored on the server.

***Decapsulation*:**

During authorization, a decapsulation algorithm is used for decapsulation and finally L and C_D are obtained. The Lvalue is regarded as seed for PRNG to acquire more indices than t-1 for retrieving the CTS_i from hash table. Finally, the user estimates CTS and Lagrange polynomial makes the user to obtain polynomials (b_{t+1}) for constructing the C_{TE} and C_{TK} and finally user tends to recover the value of C_T.

***Data Decryption*:**

If the user acquires the data requested in the encrypted form, decryption is used to retrieve the original data using recursive algorithm Decrypt(C_T,S_K,x) based on three inputs C_T, S_K and x node from T.

4.4. SECURITY ANALYSIS

This section analysis the security of the proposed scheme and it shows how the proposed scheme meets the security definitions.

The proposed scheme is selective security if all polynomial time adversaries have at most a negligible advantage.

Initialization Phase:

The adversary A tends to submit the challenged access structure A to the challenger C.

Setup Phase:

The challenger C runs the setup algorithm and sends the public parameters PP to the adversary A and keeps the master key MSK to itself.

Encryption Phase:

The adversary A adaptively issues repeated secret keys corresponding to attribute sets, where none of these attribute sets satisfy the access structure A

Challenge Phase:

The adversary A submits two equal-length messages M0 and M1 to C. The challenger C randomly selects a bit $b \in \{0,1\}$ and encrypts the message Mb for the access structure A . The challenger C sends the ciphertext CT to the adversary A.

Re-encryption Phase:

Encryption is repeated. The adversary A outputs a guess b_0 of b. If $b_0 = b$, the adversary A wins this game.

4.5. RESULT ANALYSIS

The Pairing-Based (PBC) library is used as for estimating the computational overhead of the proposed method. The experiments are conducted on a desktop computer with Intel corei7 3.4 GHz processor having 500GB storage capacity on an 8GB RAM. The computations are carried out for several iterations and the average values are reported. The experiments are conducted with different file sizes that ranging between 20 – 240 kilobytes.

The computational overhead is estimated in terms of data encryption overhead, extraction of ciphertext, decryption key overhead, shares generation overhead for the ciphertext and ciphertext shares distribution overhead. The computational overhead for different file size is given in Figure 4.2.

Table 4.1: Computational overhead for different file size

File Size (kB)	TopK	SEED	Attribute Based Encryption
1	120	99	90
2000	155	134	125
4000	190	169	160
6000	225	199	190
8000	260	229	220
10000	295	259	250

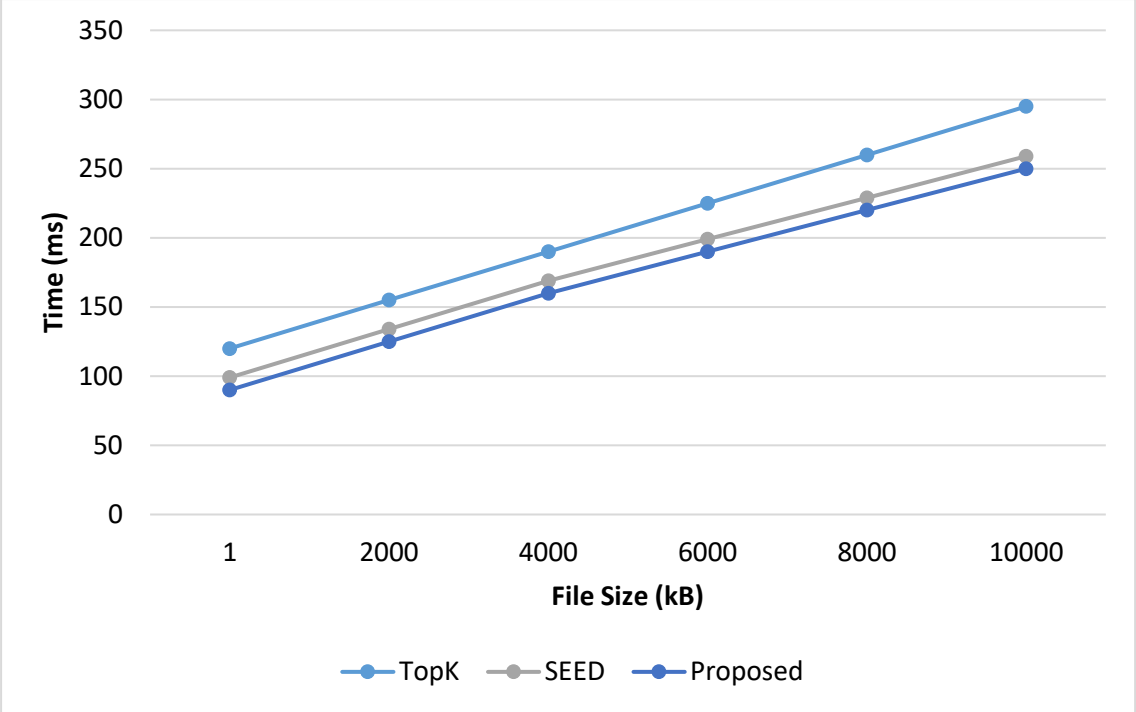


Figure 4.2: Computational overhead for different file size

It is seen that the overhead of the existing methods are higher than the proposed method in all these operations. The increase in overhead in the existing methods is due to the association of cipher text prior the extraction of cipher text parts. On other hand, non-association with cipher text at the time of its extraction leads to lower computational overhead. The overhead in all the methods including the proposed method seems linear with respect to increasing execution time and file size.

Table 4.2: Encryption time with different attribute size

File Size (kB)	TopK	SEED	Attribute Based Encryption
1	122	50	42
10	277	200	187
20	389	270	244
30	494	400	365
40	603	500	463
50	800	620	593

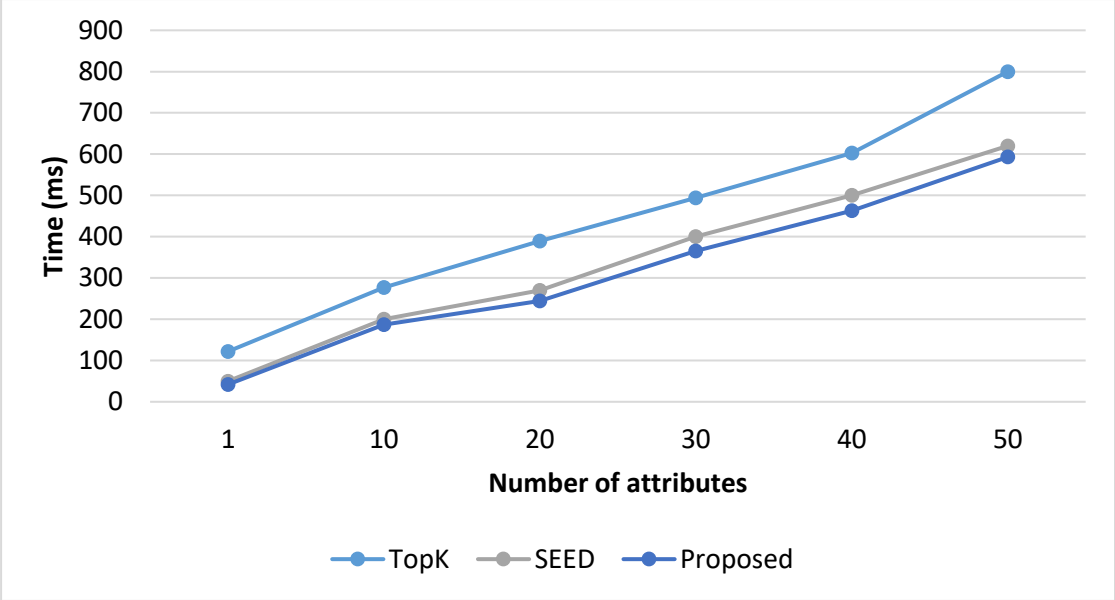


Figure 4.3: Encryption time with different attribute size

The results of encryption time is given in Figure 4.3 and Table 4.2, the results of decryption time is given in Figure 4.4 and Table 4.3 With respect to varying user attributes. The comparison is made between the proposed and the existing methods and the evaluation is made in milliseconds. The varying user attributes between 1 and 50 shows that the results are linear with increasing user attributes. The simulation result shows that the time for encryption in the proposed method is lesser than the existing methods. Similarly, the time for decryption in the proposed method is lesser than the existing methods.

Table 4.3: Decryption time with different attribute size

File Size (kB)	TopK	SEED	Attribute Based Encryption
1	100	44	23
10	250	190	190
20	322	272	250
30	458	358	332
40	588	449	435
50	683	632	603

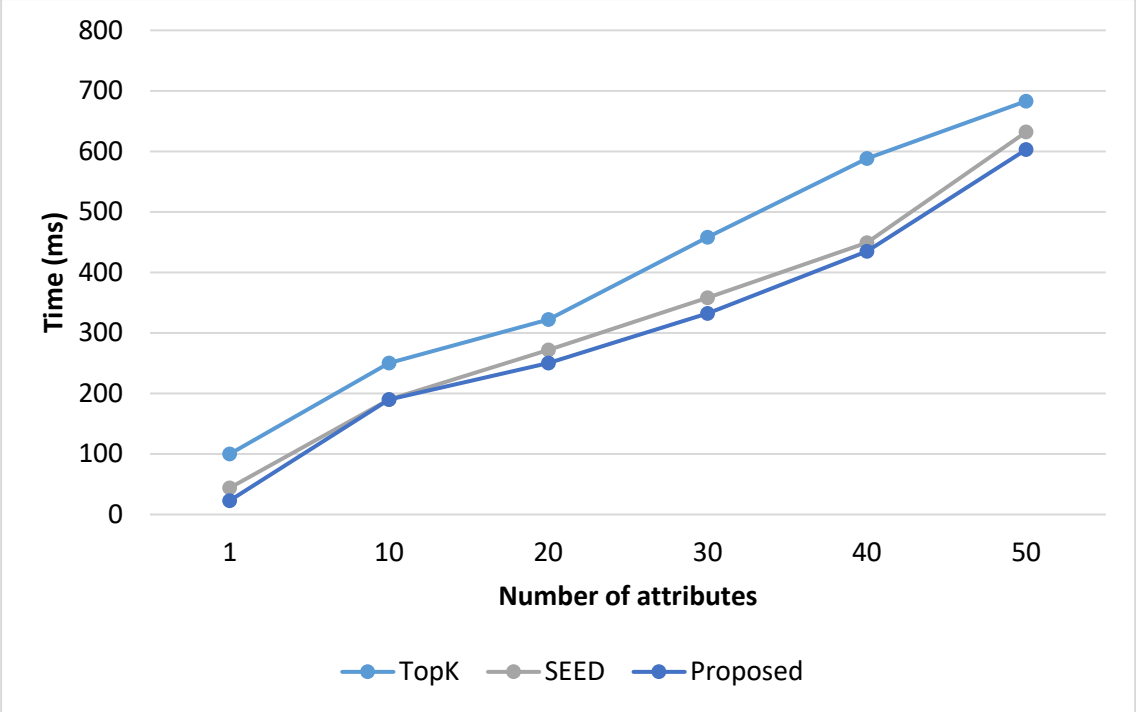


Figure 4.4: Decryption time with different attribute size

The results of encryption time is given in Figure 4.5 or Table 4.4 and the results of decryption time is given in Figure 4.6 With respect tovarying file sizes. The comparison is made between the proposed and the existing methods and the evaluation is made in milliseconds.

Table 4.4: Encryption time with different file size

File Size (kB)	TopK	SEED	Attribute Based Encryption
1	1100	700	100
100	1500	1250	740
200	1900	1700	1323
300	2200	2000	1743
400	2423	2300	2239
500	2746	2500	2493

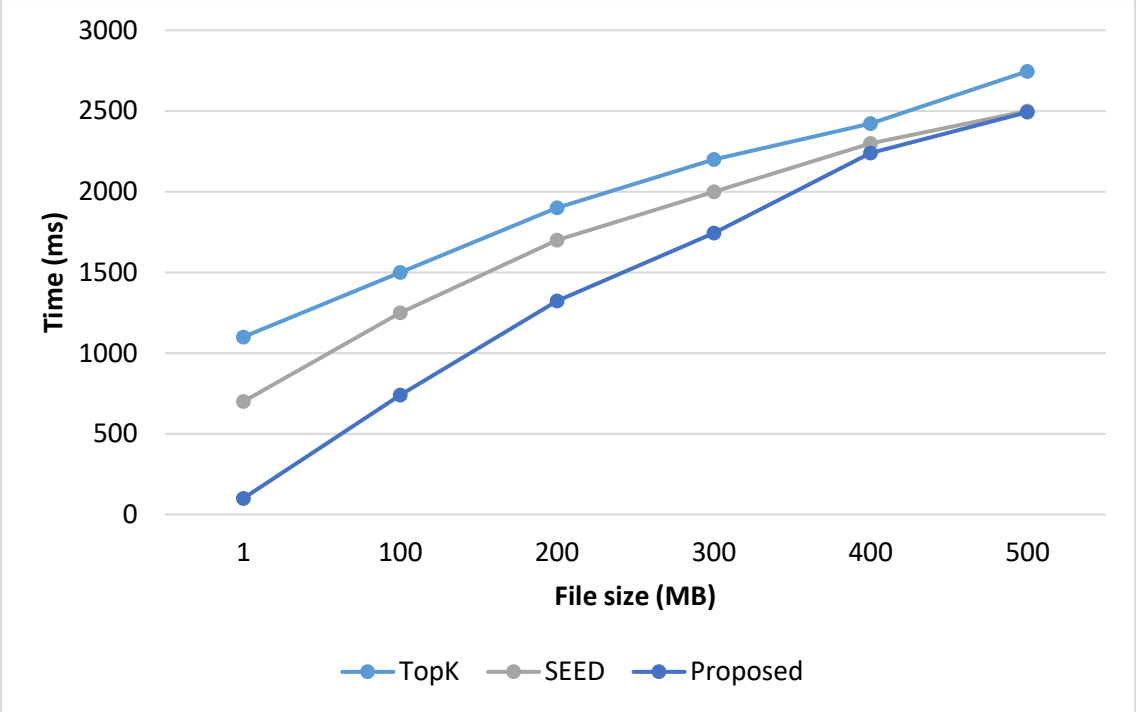


Figure 4.5: Encryption time with different file size

Table 4.5: Decryption time with different file size

File Size (kB)	TopK	SEED	Attribute Based Encryption
1	1023	600	250
100	1423	1002	734
200	1966	1432	1233
300	2342	1765	1734
400	2954	2303	2102
500	3124	2643	2543

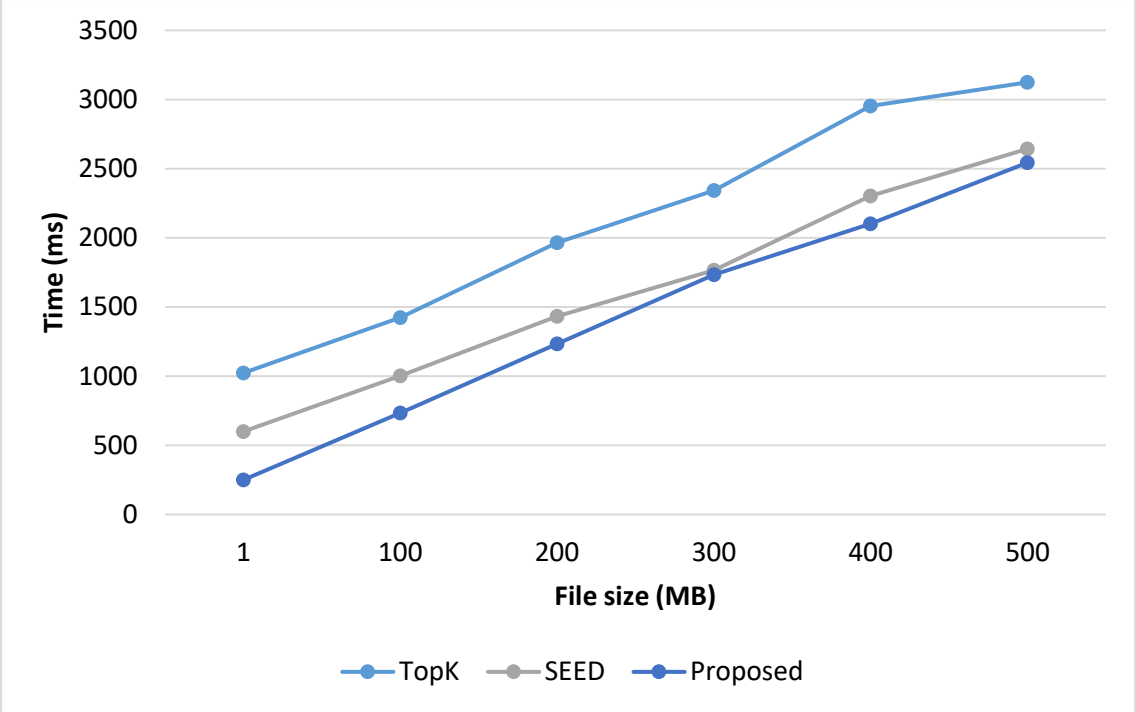


Figure 4.6: Decryption time with different file size

The varying file size between 1 and 500 MB shows that the results are linear and then it begins to record a constant growth with increasing file size. This condition is the same for both encryption and decryption time, since the proposed method uses public key cryptography. The simulation result shows that the time for encryption in the proposed method is lesser than the existing methods for varying file size. Similarly, the time for decryption in the proposed method is lesser than the existing methods for varying file size.

4.6 CONCLUSION:

In this chapter, a state-of-art serverless scheme is introduced in this chapter using an effective security mechanism that secures the data in serverless computing environment. The secured data in serverless environment improves the knowledge and resource sharing and protects the data against many attacks. The attributes of the user deployed in this scheme improves the serverless security and improves the performance against PBC library. The simulation result shows that the proposed method is effective for providing security against the resource or data before it gets accessed by the third party.

CHAPTER -V

FILTER BASED APPROACH FOR CLOUD SERVERLESS ENVIRONMENT

CHAPTER 5

FILTER BASED APPROACH FOR CLOUD SERVERLESS ENVIRONMENT

5.1. INTRODUCTION

When VM's are randomly deployed on the network, serverless computing, node authentication and message reliability are the main problems [136]. These issues lead to poor network performance and therefore serverless computers do not operate on a regular basis [137].

Attackers try attacking the intermediate VM on the serverless computer during the server maintenance phase. Thus, data transmission may not be possible due to the involvement of the attackers. Additionally, the extraction and tunneling of data from VMs involves other VMs. It modifies the message or data content of attackers during the data transmission phase. Finally, the original data is manipulated by users and attackers of third parties [138].

Several algorithms like Harris Corner Optimization based image retrieval [139], Trustworthy agent-based encrypted access control method [140], hybrid encryption algorithm [141], ID-based authentication [142], Homomorphic encryption [143], Public-Key Encryption [144], attribute-based encryption scheme with policy update and file update algorithms [145] are being used to resolve these problems and mitigate the intrusion of attackers. This secure protocol makes serverless computing reliable for transmitting and receiving high-level authenticity messages.

There are various ways to reduce active and passive serverless attacks [146]. Several solutions, like detection systems or protocols and prevention methodologies, are also used for preventing attackers from behaving in a malicious manner. These protocols focus mainly on the arrival rate and data delivery ratio of data transmission.

Furthermore, the anonymous protocol identifies the misrepresentation in the process of route discovery and services between VM's in serverless computing, namely anonymous routing request as well as non-request protocol [147]. However, the performance of the serverless computer degrades at the time of regular maintenance, making it difficult to identify the attackers involved in the transmission of data. This causes duplicate VM data to be transmitted in the serverless computer to the respective destination VM.

Therefore, the idea of encryption is used to distribute secret or key information and generate or decrypt signatures among different parties, in a single way in order to prevent misuse or failure. Signature encryption is a helpful way to decentralize the power to send a message on the serverless computer [148]. The Signature Protocol scheme allows a subset of players to create the signature. If less players participate in a protocol, the valid signature will not be generated [149].

The public key cryptosystem idea uses the public key from the VM identity, for example. IP or e-mail address. A third-party or a trusted private key generator generates the public key and passes the key on to an adequate user via a secured communications channel. The main downside is that no single string recognizes users based on the attributes for each other [150].

A Fuzzy based encryption [142] is applied to improve the process of a unique string for each user, which describes the identity of the user. This prevents the idea of a single string. For the privately-held user, a cypher text can be decrypted which can be encrypted through the attributes set if the attributes overlap. Biometric identification is the most common application used for users with multiple attributes, using their biometric identifiers. The information is biometrically encrypted which is called attribute-based encryption.

The intrusion is avoided in existing techniques by discarding VM's, which are malicious in the serverless computer. The malicious VM's creates an intolerable link in serverless computer, which results in huge data loss leading to server free computer congestions in a wider range. The reliability of the connection is therefore considered an important factor during system design.

5.2. PROPOSED ENCRYPTED FUZZY BASED FILTER

The research proposed includes the use of a fuzzy based signature authentication system using fuzzy concepts to enhance authentication of data in VMs. The attacker is eliminated and all the vulnerabilities resulting from malware are removed. This is generally eliminated by evaluating the reliability of the link before route maintenance, which guarantees better data supply and thus starts data transmission. The serverless computing connection reliability helps to improve the serverless computing performance. After the data transmission, residual energy is used to assess the efficiency of the fuzzy authentication scheme. With the help of fuzzy-based process, the serverless computing cost of the proposed system is reduced and data integrity is maintained by enhanced signature verification.

The method is implemented in a non-centralized server-free environment in which VMs without an administrator are not physically connected. The routes are calculated using serverless computing on the basis of the source VM (S) request to transport the data to the destination VM (D). The calculated neighbour VMs are used to transmit the information from source to destination VM. The query list of the route requests is used to transmit the data to the destination VM. The serverless computing topology is based on VM stability, stabilization of links, residual energy and signature checks.

5.2.1. LINK RELIABILITY ESTIMATION

The connection quality is used to assess the connection reliability and is defined as the signal-to-noise relationship of the connection. When the Signal to Noise Rate is high, the Bit Error Rate (BER) value is low, where SNR is inversely a promotional BER. The lower mobility rate improves the connection reliability in the serverless computer. The next VMs can be found in the following relationship,

$$SNR_{in} = \chi \cdot SNR_{in} + (1 - \chi) SNR_{avg} + L_{st} \quad \textbf{(5.1)}$$

Where,

SNR_{in} is the current SNR value,

SNR_{avg} is the average SNR value over a period of time *t*.

χ - constant that lies in the range [0,1].

From the above discussion, the value of the current SNR is also found to be more. Also, when the threshold value for SNR is reduced compared to current SNR, the quality of the connection is assigned with value one and vice versa.

5.2.2. VM RELIABILITY ESTIMATION

The reliability of VM in the serverless computing is obtained using stability of VM and expiration time of the link. The reliability of VM depends on the stability of each VMs in serverless computing, $N_s(\tau)$ with an expiration of link or connection $T_{LE}(\tau)$.

Thus, the factor for reliability estimation for a VM, $N_{rf}(\tau)$ is estimated as follows:

$$N_{rf}(\tau) = f\left(N_s(\tau), T_{LE}(\tau)\right) \quad \textbf{(5.2)}$$

If stability > 0 then the VM reliability is estimated which reduce the routes failure in serverless computing using VMs with an adjustment topology.

5.2.3. RESIDUAL ENERGY FACTOR ESTIMATION

The main factor in determining the efficiency of server free computing is energy consumption for each process. Using energy management systems that increase residual VM energy, overuse is avoided. Hence, the residual energy factor of the VM $R_{et}(\tau)$ is estimated as,

$$R_{et}(\tau) = \frac{E_{rm}(\tau) - E_{pl}(\tau)}{E_T(\tau)} \quad \textbf{(5.3)}$$

Where,

$E_{rm}(\tau)$ is the average energy spent in each VMs over a time period (τ).

$E_{pl}(\tau)$ is the energy loss in each VMs during a time period (τ), and

$E_T(\tau)$ is the energy distribution over VMs over a time period, τ.

5.3. FIS IMPLEMENTATION

The proposed FIS uses a sugeno type-2 model for the detection of malicious actions in VMs in serverless computing, which is illustrated in Figure 5.1.

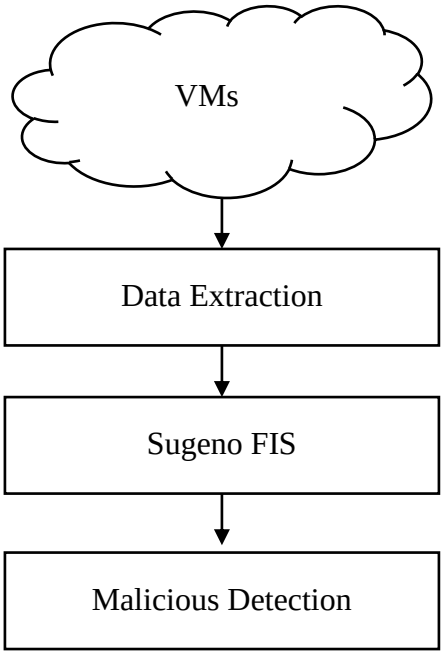


Figure 5.1. Proposed Fuzzy Architecture

5.3.1. FUZZIFICATION

As discussed above, the fuzzy input system is loaded with three input factors. The number of reliable links is defined by low factor, medium factor reliability of VM and high factor input in residual energy as shown in Figure 5.2.

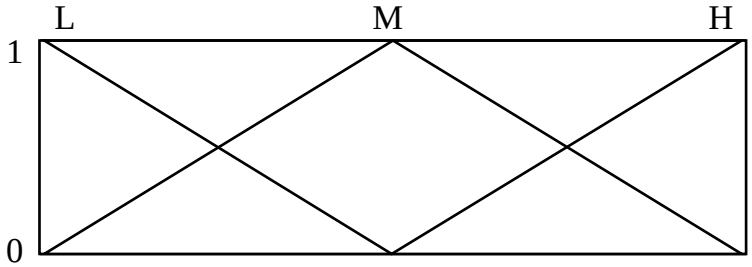


Figure 5.2. Fuzzy membership function for the proposed system

With references to the outputs, the output variable in regards with the fuzzy logic system is estimated and that ranges between extremely low, low, medium, high and extremely high as shown in Figure 5.3.

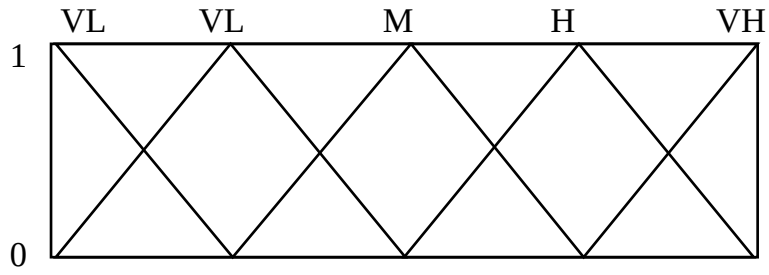


Figure 5.3. Cost value of Fuzzy membership functions

The FIS is deployed with IF-THEN rules to achieve the required output from the respective input. Reduced costs can be achieved through maximum connection, VM reliability and maximum residual energy, which are the favourable inputs of the fuzzy system. The rule of the system proposed is based on the following rule:

If R_l (medium) **AND** R_n (low) **AND** low E_r, then the output is considered to be very low.

Defuzzification: The obtained value from FIS helps the Defuzzification process to find the trust level of each messages transmitted to VMs, which is based on the input variables (R_n, R_l, E_r).

5.3.2. FUZZY ANALYZER

Furthermore, the trust mechanism represents the reliability of the VM, where the positive and negative experience is the inclination and decline of the levels of trust. The flippant logic helps effectively deal with inaccuracy and uncertainty. The confidence-based fuzzy logic calculates the confidence value of the VM. The trust values are based on R_n, R_l, E_r and used as a foggy input, which predicts that the VM will be malicious or not. The certificate authority requesting the VMs to exchange data using certificates. Here, a furious logic confidence-based algorithm is used by verification of certificates to perform the task of data exchange. The fuzzy-based analyzer is designed with trust based value which exchanges data using a critical threshold value based on marking the VMs as malicious or non-malicious. When the threshold value is lesser than the critical threshold, the VM is isolated and vice versa. Table 5.1 shows the confidence values of the VMs with fuzzy logic.

Table 5.1 Fuzzy Discrimination

Level	Fuzzy levels	Semantics	Trust Values
5	Very high	Highly not infected	0.8 to 1
4	High	Likely not infected	0.6 to 0.8
3	Medium	Partially not infected	0.4 to 0.6
2	Low	Likely infected	0.2 to 0.4
1	Very low	Highly likely infected	0 to 0.2

5.3.3. IMPLEMENTATION METHOD

The proposed method is used to identify internal attacks such as attacks flooding, DoS attack, congestion, resource expenditure and bandwidth exhaust. The proposed fuzzy monitoring in serverless computing uses several parameters that includes sequence number of route request rate, charging pattern for Fractional Flow Reserve (FFR) attack detection and recognition time.

The core architecture of the method proposed has four steps:

- Collection of log file from each VMs
- Analysis
- Evaluation
- Response

The computer decision without the server is based exclusively on the Hacking level, which is estimated by combining the total number of time and RREQ. The global response and local reaction module are responsible for local and global reactions, and the reaction is sent to every neighboring VM in the global response.

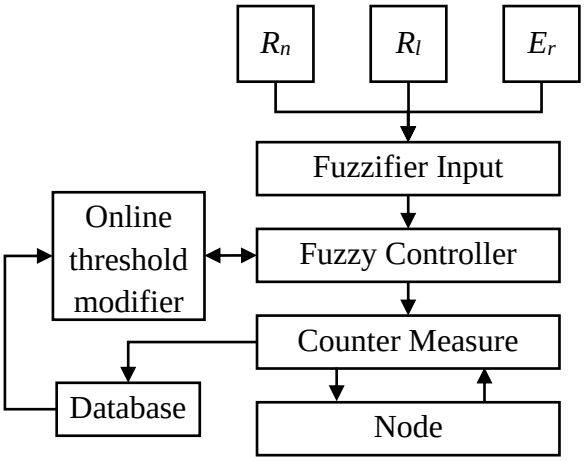


Figure 5.4. Proposed encrypted fuzzy based filter

The rules for fuzzy interference system is categorized based on trust values, which is given as follows:

1. If trust value is 5, then the data entered in VM is non-malicious (it is highly not infected)
2. If trust value is 4, then the data entered in VM is non-malicious (it is likely not infected)
3. If trust value is 3, then the data entered in VM is non-malicious (it is partially not infected)
4. If trust value is 2, then the data entered in VM is Malicious (it is likely infected)
5. If trust value is 1, then the data entered in VM is Malicious (it is highly infected)

The proposed method is then used to check the confidence value of the requested VM and the fuzzy table is then updated with the confidence value using the lookup function. The VM is regarded as malicious or not malicious depending on the results of the algorithm. Malicious VMs are removed in order to avoid breakdown of the link from serverless systems or isolated from the serverless computer.

5.4. SIGNATURE VERIFICATION SCHEME

The proposed protocol uses the concept of a Probably Secure Encryption Curve (PSEC) (Okamoto, T., et al. 2000) to improve the data integrity in serverless computing. This ensures better security for VMs in serverless computing with PSEC key encapsulation scheme. The first elements of PSEC are shown in Table 5.2:

Table 5.2. Notations

Primitives	Description
KDF	Key derivation Hash function
MAC	Message authentication code
AES	Symmetric-key encryption
DEC	Decryption
ENC	Encryption

5.4.1. Proposed Encryption Algorithm

Input: Verification of domain parameter $T = (E, FR, G, x, y, V, n, h)$ considering a plain message text (m) with a public key U.

Output: Obtaining Cipher text (R, C, s, t).

Selection of a variable $y \in Y \{0,1\}^l$, where l is regarded as the length of bit n.

Estimate the Function for Key Derivation

$$(\boldsymbol{k, k_1, k_2}) \leftarrow \boldsymbol{KDF(y)} \qquad \textbf{(5.4)}$$

where $k = 128 + l$, which is a bit length

Estimate $k = k \bmod n$.

Estimate $Y = kV$ and $Z = kE$.

Find $s = y \oplus KDF(Y, Z)$

Find $C = ENCk_1(m)$ and $t = MACk_2(C)$

Return (Y, C, s, t)

5.4.2. Proposed Decryption Algorithm

Input: Verification of domain parameter $D = (e, FR, G, x, y, V, n, h)$ considering a private key d with a cipher text (Y, C, s, t).

Output: Plaintext m tends to accept the cipher text.

Estimate $Z = dY$.

Estimate $y = s \oplus KDF(Y, Z)$.

$$(k, k_1, k_2) \leftarrow KDF(y) \text{ with } k \text{ bit length} = l + 128 \qquad (5.5)$$

Estimate $k = k \bmod n$.

Estimate $R_1 = kV$.

If $Y_1 = Y$ then the cipher text is discarded

Estimate $t = MACk_2(C)$

If $t = 1$ then the cipher text is discarded.

Decrypt the message using $m = DECk_1(C)$

Return(m)

Table 5.3. Data format

Source ID	Destination ID	Link and VM reliability	Hop Count	CRC	Residual energy
2	2	4	1	2	4

Table 5.3 shows the data format for transmitting the data, where the source VM and classification VM contained 2 bytes. The hop count filed is 1 byte and determines the total VM(s) associated with a particular VM in the cluster. The four bytes of connection and vector retirement shows the decrease in VM and the reliability of the vector threshold value. During road maintenance, the residual energy determines the power left in the VM. The last field is filled with 2 bytes of Cyclic Redundancy Check (CRC) used to correct and detect errors.

5.5. RESULT ANALYSIS

The proposed method is carried out with the NS-2 simulator and the protocol is tested and evaluated. The whole mobile VMs with 50 ad hoc VMs move randomly at low speed in 1200 m^2 for 75 seconds in the proposed simulation. In a serverless computing, the transmission range of VMs is considered to be 250 m and the simulation traffic is constant bit rate (CBR). Table 5.4 shows the complete simulation settings.

Table 5.4 Simulation settings and parameters

Parameters	Value
Area Size	1200 × 1200
Transmission Range	250m
Total number of VMs	50
MAC layer	802.11
Operating Frequency	2.4GHz
Total Simulation Time	75 sec
Maximum connection	10
Type of attack	DDoS
Data Size	512 bytes
Propagation mode	Free space
Mobility Model	Random Way Point
Traffic Source	CBR (UDP)
Simulation time	75s
Movement speed	1 ms^{-1}

The proposed metric is evaluated using following metrics, which is shown below:

***End to end Delay*:**

The end-to-end delay is defined as the delay in transmitting data from source to destination VM.

***Control Overhead*:**

The control overhead is defined as the ratio between the total number of data received and the total number of data sent.

***Data Reliability*:**

Data reliability is defined as the ratio of the total number of original data received to the total number of original data transmitted.

***VM Reliability*:**

The reliability of VM is defined as genuine VM which cannot be negotiated with a faulty VM.

***Data Integrity*:**

Data integrity is defined as modified data contents through hacked VMs.

***Residual energy*:**

Residual energy is defined as the total energy consumed after data is received.

The results are evaluated in an attacking environment against all the parameters between the proposed fuzzy-based method and conventional methods like Deduplication [132], Compressed-Encryption [151] and ADS-B security [152].

Table 5.5. Data reliability of the data transmitted

Data packets	Deduplication	Compressed-Encryption	ADS-B security	Fuzzy Encryption
200000	115646	192345	201545	215454
400000	148255	265845	324522	394255
600000	395421	445823	512426	625848
800000	521542	652214	652214	745856
1000000	594545	785265	825485	965866

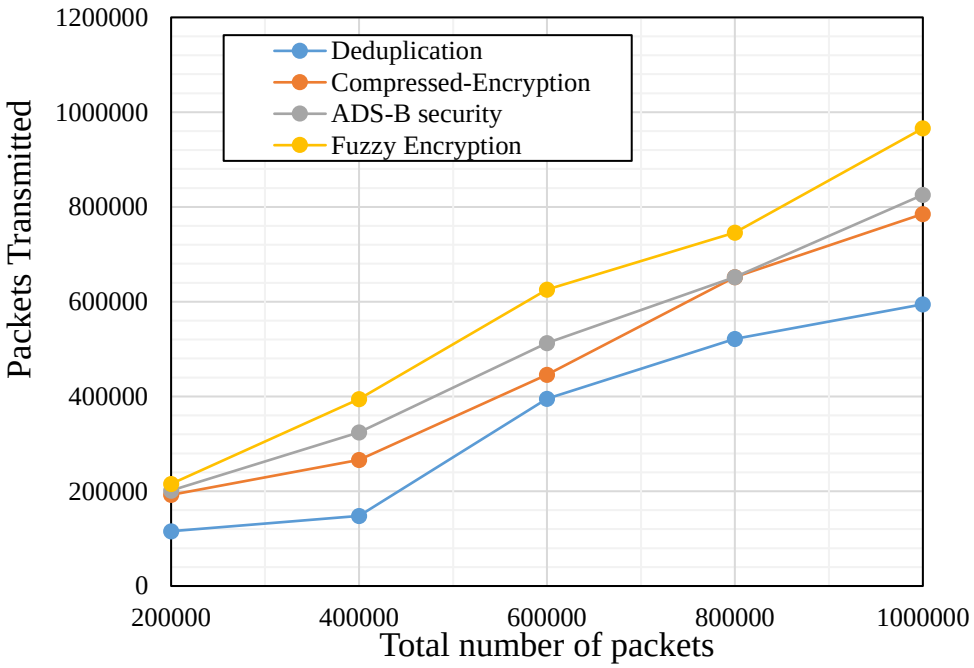


Figure 5.5. Data reliability of the data transmitted

In Table 5.5 and Figure 5.5, the reliability of the method data is shown to vary from 0 to 100 and the proposed method is shown to be more data reliable than other conventional protocols. This is because the proposed method is present, whereby each VM is evaluated at all times compared to other algorithms.

Table 5.6. Control overhead of the VMs in the serverless computing

Data packets	Deduplication	Compressed-Encryption	ADS-B security	Fuzzy Encryption
200000	6500000	5600000	4750000	3458454
400000	5400000	5200000	4455000	3345484
600000	3600000	3200000	2895656	2487875
800000	2400000	2200000	1758484	1144555
1000000	2600000	1600000	1258899	754644

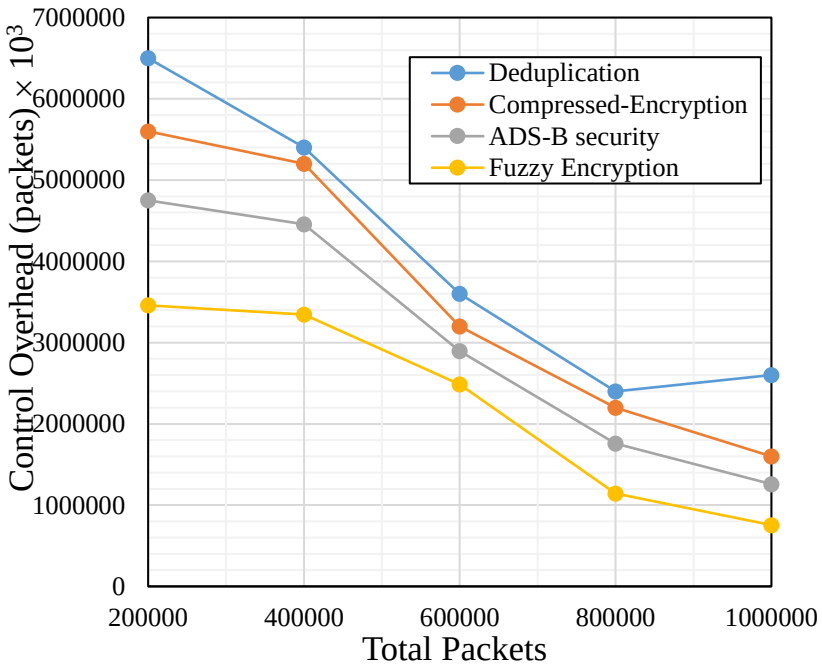


Figure 5.6. Control overhead of the VMs in the serverless computing

Table 5.6 and Figure 5.6 shows the overall comparison of the control overhead because of the serverless multiple dispatch of control messages and compared with VM mobility. It is apparent from the results that the proposed method achieves low overhead control because the control messages sent by VM are small, and the overhead decreases when mobility is slow. This does not apply to other conventional arrangements.

Table 5.7. End to End delay of the VMs in the serverless computing

Data packets	Deduplication	Compressed-Encryption	ADS-B security	Fuzzy Encryption
200000	1654548	2011452	2954656	3625242
400000	2105545	2548478	3645485	4500002
600000	3484625	4025235	4854652	5244252
800000	3654852	4000000	5245835	6254823
1000000	3955463	4525222	6258465	7154228

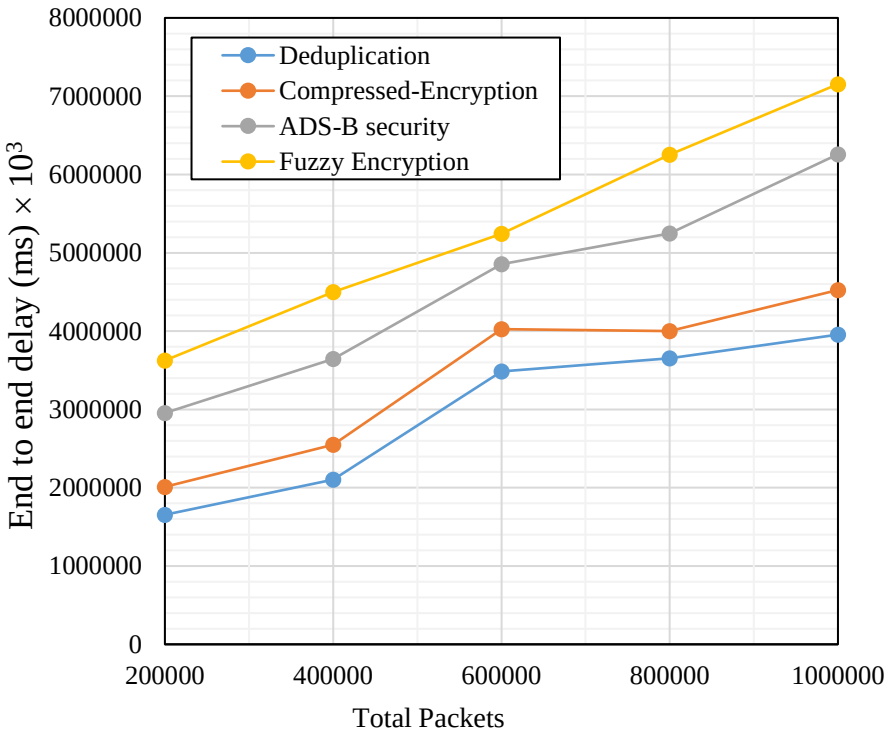
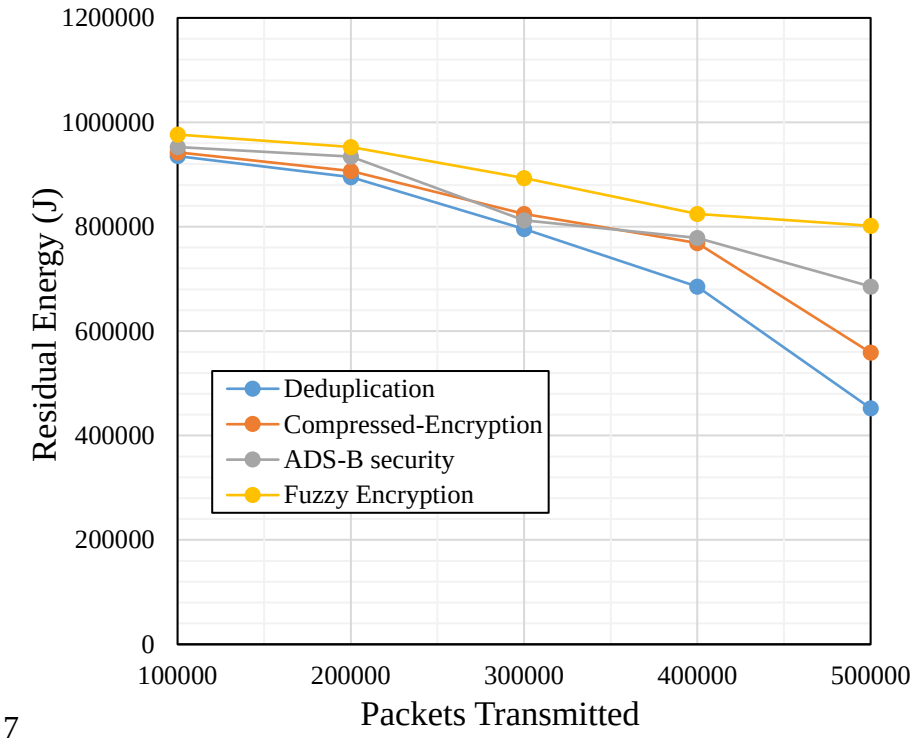


Figure 5.7. End to End delay of the VMs in the serverless computing

The results of the average final delay between the proposed model and other conventional models against VM mobility are shown in Table 5.7 and Figure 5.7. The results show that the method proposed is less late than conventional. The high results are due to lower overhead control, greater VM reliability and reliability. The results are very efficient.

Table 5.8: Residual energy of the VMs in the serverless computing

Data packets	Deduplication	Compressed-Encryption	ADS-B security	Fuzzy Encryption
100000	935254	942544	952658	976526
200000	895325	906542	934548	952458
300000	795822	824548	812448	893454
400000	685423	768542	778454	824545
500000	452151	558787	685547	801645



7

Figure 5.8. Residual energy of the VMs in the serverless computing

The residual energy comparison of the proposed and current methods, which varies from 10 to 50ms, is presented in Table 5.8 and Figure 5.8. The result is that the system proposed achieves greater residual energy than the methods already in use. This is because of the lower calculations and the reduced serverless computing control overhead than conventional techniques.

Table 5.9: VM reliability

Data packets	Deduplication	Compressed-Encryption	ADS-B security	Fuzzy Encryption
200000	128554	146585	162548	201255
400000	215222	302122	325245	335245
600000	384542	495852	514554	596856
800000	495245	625353	694522	742536
1000000	526586	735954	824548	924877

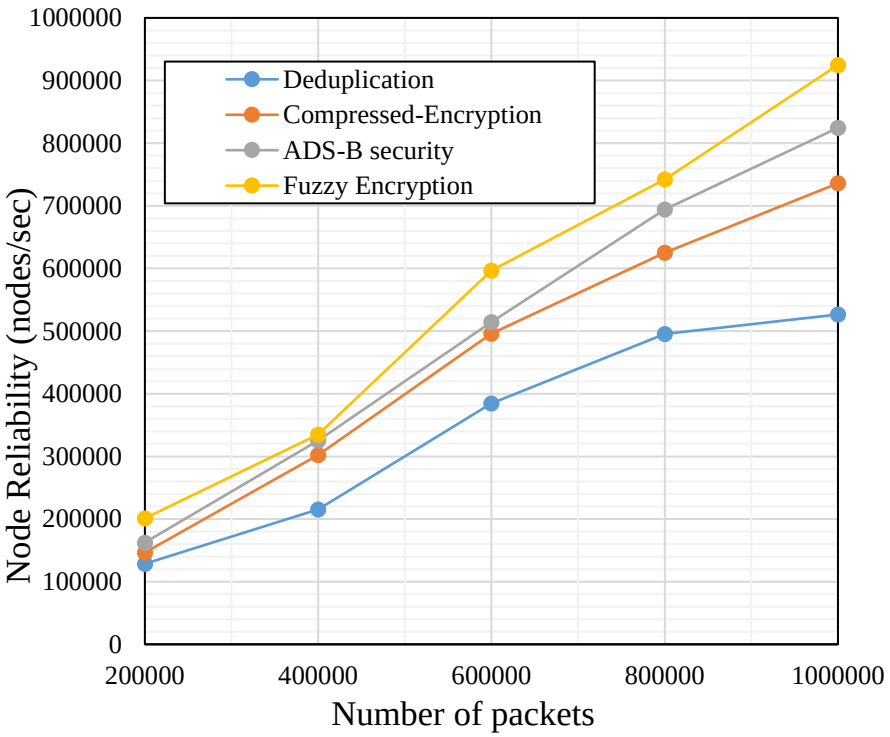


Figure 5.9. VM reliability

The reliability of VMs With respect to the number of VM in serverless computing is shown in Table 5.9 and Figure 5.9, the results show that the proposed method has a higher reliability than conventional methods.

Table 5.10: Data Integrity

Data packets	Deduplication	Compressed-Encryption	ADS-B security	Fuzzy Encryption
100000	521452	565484	592454	925482
200000	1024522	1358550	1785482	1925834
300000	1852248	2235325	2485268	2745623
400000	2485246	3584552	3625522	3824586
500000	2865853	3754842	3968452	4623258

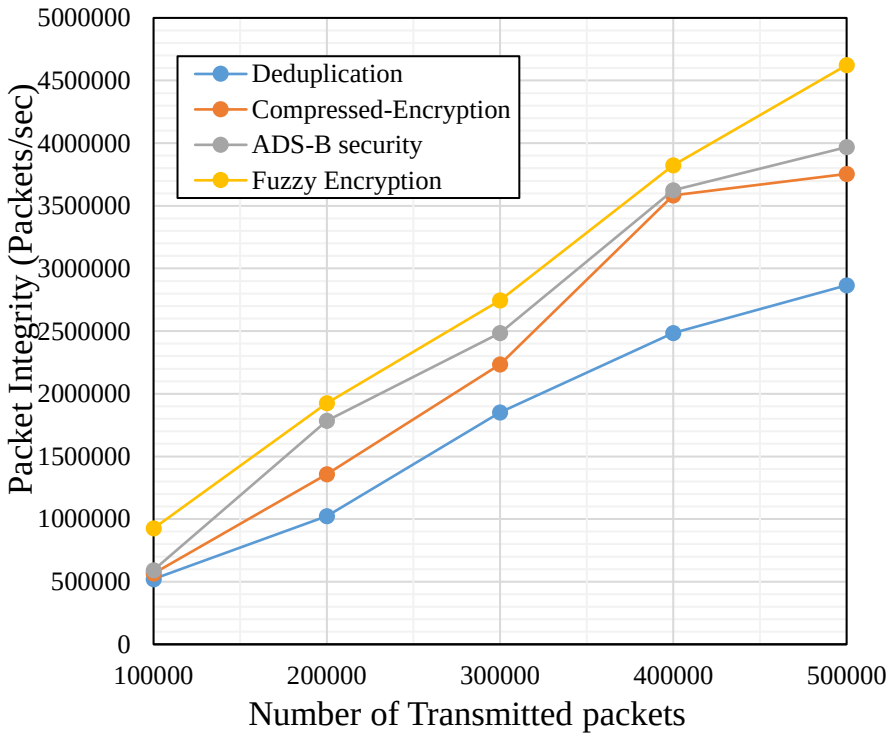


Figure 5.10. Data Integrity

Table 5.10 and Figure 5.10 shows the integrity of the data when the data is transferred between the VMs. The results show that the proposed method achieve high data integrity using the proposed method in terms of encryption and decryption compared to the other methods.

This chapter focus mainly on eliminating the attacks while transmitting and ensure better delivery of data to the destination VM. The proposed method is tested and proved to be efficient in terms of high data integrity, improved residual energy and VM reliability. In addition, a fuzzy model reduces calculation costs and increases malicious VM detection in serverless computing. With these calculations, the VMs are verified and data are thus efficiently transmitted.

This codes the authentication, integrity and confidentiality problems in serverless computing with user authentication of information in serverless computing with the encrypted Fuzzy Logic filter. The simulation is performed to verify the performance of the other safety models. The results show that the proposed fuzzy filter mechanism achieves better performance with higher overhead compared to existing methods. The result proves better in highspeed data transmission between VM and ensures better security and reliable transmission of data than the other methods.

CHAPTER -VI

PERFORMANCE ANALYSIS

CHAPTER 6

PERFORMANCE ANALYSIS

6.1. INTRODUCTION

This chapter conducts an experimental analysis between all the three proposed models say 1) Identity based cryptosystem 2) Attribute based encryption 3) Filter based security. The experimental analysis is conducted in terms of different security metrics that includes: encoded key size, signature time and key generation time in terms of its minimum, average and maximum time. Further, it is tested in terms of serverless network metrics that includes delay time, average throughput, average response time, error rate, load distribution and cost-efficiency.

6.2. PERFORMANCE ANALYSIS OF IDENTITY BASED CRYPTOSYSTEM

This section provides the experimental analysis on Identity based cryptosystem. The experimental analysis is conducted in terms of different security metrics that includes: encoded key size (Table 6.1), signature time (Table 6.2) and key generation time (Table 6.3) in terms of its minimum, average and maximum time. Further, it is tested in terms of serverless network metrics that includes delay time (Table 6.4), average throughput (Table 6.4), average response time (Table 6.4), error rate (Table 6.4), load distribution (Table 6.5) and cost-efficiency (Table 6.5).

Table 6.1: Encoded Key Size

Key size (bits)	Encoded Key size (bytes)
128	112
256	126
320	142
384	160
512	180
576	202

640	228
704	256
768	289
832	325
896	366
960	412
1024	425

From the Table 6.1, it is seen that with increasing key size and with varying key size, the performance of model is not affected since, the encoded key size is lesser than the original key size in Identity based cryptosystem.

Table 6.2: Signature Time

Key size	Signature Time
128	0.121
256	0.150
320	0.186
384	0.231
512	0.286
576	0.355
640	0.440
704	0.545
768	0.676
832	0.839
896	1.040
960	1.290
1024	1.317

From the Table 6.2, it is seen that with increasing key size, the performance of model is not affected since the signature time is increasing at a marginal rate and not at linear rates in identity based cryptosystem.

Table 6.3: Key Generation Time (ms)

Key size (bits)	Minimum	Average	Maximum
128	9638.51	9759.67	9882.83
256	9913.87	10044.04	10149.18
320	11145.47	11257.62	11349.74
384	11282.65	11385.78	11486.91
512	11586.04	11748.25	11859.40
576	12766.58	12864.70	13001.88
640	13147.07	13300.27	13392.39
704	13510.54	13654.73	13780.89
768	13486.51	13637.71	13832.96
832	15294.86	15427.03	15523.15
896	15519.15	15661.33	15793.50
960	16214.05	16284.14	16460.37
1024	17015.55	17124.95	17252.25

From the Table 6.3, it is seen that with increasing key size, the performance of model is not affected since the key generation time in terms of minimum time, average time and maximum time is increasing at a marginal rate and not at linear rates in identity based cryptosystem. It is seen that that the average time is the iterations of all rounds between the minimum time and the maximum time.

Table 6.4: Network Performance Metrics

Metrics	Key size (bits)	
	128 (minimum)	**1024 (maximum)**
Key size (bits)	120.25s	165.5s
Delay Time	541 ms	592 ms
Average Throughput	78.6 hits/second	79.2 hits/second
Average Response time	54.3 ms	52.32 ms
Error Rate	0.23 %	0.29 %

The Table 6.4 shows reduced error rate in least percentages, reduced delay in milliseconds and there is a marginal difference between the minimum bits and maximum bits. Also, the average throughput is not a variable one and it maintains at constant rate for the minimum and maximum bit.

Table 6.5: Load vs. Cost

Load (VMs)	Cost (USD)
5	2.51
10	3.13
15	3.91
20	4.88
25	6.09
30	7.60
35	9.48
40	11.84
45	14.77
50	15.26

The Table 6.5 shows cost of VMs for resource allocation and storage for allocating higher rate of data. It is seen that allocation of increasing VMs tends to increase the cost at marginal rate.

6.3. PERFORMANCE ANALYSIS OF ATTRIBUTE BASED ENCRYPTION

This section provides the experimental analysis on Attribute based encryption. The experimental analysis is conducted in terms of different security metrics that includes: encoded key size, signature time and key generation time in terms of its minimum, average and maximum time. Further, it is tested in terms of serverless network metrics that includes delay time, average throughput, average response time, error rate, load distribution and cost-efficiency.

Table 6.6: Encoded Key Size

Key size (bits)	Encoded Key size (bytes)
128	111
256	125
320	140
384	158
512	178
576	200
640	225
704	254
768	285
832	321
896	362
960	407
1024	420

From the Table 6.6, it is seen that with increasing key size and with varying key size, the performance of model is not affected since, the encoded key size is lesser than the original key size in Attribute based cryptosystem.

Table 6.7: Signature Time

Key size (bits)	Signature Time (ms)
128	0.120
256	0.148
320	0.184
384	0.228
512	0.283
576	0.351
640	0.435
704	0.539
768	0.669

832	0.829
896	1.028
960	1.275
1024	1.302

From the Table 6.7, it is seen that with increasing key size, the performance of model is not affected since the signature time is increasing at a marginal rate and not at linear rates in attribute based encryption.

Table 6.8: Key Generation Time (ms)

Key size (bits)	Minimum	Average	Maximum
128	9528.93	9648.71	9770.47
256	9801.16	9929.85	10033.79
320	11018.75	11129.63	11220.70
384	11154.37	11256.33	11356.32
512	11454.32	11614.68	11724.56
576	12621.43	12718.44	12854.06
640	12997.60	13149.05	13240.13
704	13356.94	13499.48	13624.21
768	13333.18	13482.66	13675.69
832	15120.97	15251.64	15346.67
896	15342.71	15483.28	15613.94
960	16029.71	16099.00	16273.23
1024	16822.10	16930.25	17056.10

From the Table 6.8, it is seen that with increasing key size, the performance of model is not affected since the key generation time in terms of minimum time, average time and maximum time is increasing at a marginal rate and not at linear rates in attribute based cryptosystem. It is seen that that the average time is the iterations of all rounds between the minimum time and the maximum time.

Table 6.9: Network Performance Metrics

Metrics	Key size (bits)	
	128 (minimum)	**1024 (maximum)**
Key size (bits)	119.15s	162.88s
Delay Time	537 ms	589 ms
Average Throughput	79.2Kbps	79.9Kbps
Average Response time	53.6 ms	51.30 ms
Error Rate	0.21 %	0.27 %

The Table 6.9 shows reduced error rate in least percentages, reduced delay in milliseconds and there is a marginal difference between the minimum bits and maximum bits. Also, the average throughput is not a variable one and it maintains at constant rate for the minimum and maximum bit.

Table 6.10: Load vs. Cost

Load (VMs)	Cost (USD)
5	2.48
10	3.10
15	3.86
20	4.82
25	6.02
30	7.51
35	9.38
40	11.70
45	14.60
50	15.09

The Table 6.10 shows cost of VMs for resource allocation and storage for allocating higher rate of data. It is seen that allocation of increasing VMs tends to increase the cost at marginal rate.

6.4. PERFORMANCE ANALYSIS OF FUZZY FILTER BASED SECURITY

This section provides the experimental analysis on fuzzy filter based security. The experimental analysis is conducted in terms of different security metrics that includes: encoded key size, signature time and key generation time in terms of its minimum, average and maximum time. It is also tested in terms of network metrics that includes delay time, average throughput, average response time, error rate, load distribution and cost-efficiency.

Table 6.11: Encoded Key Size

Key size (bits)	Encoded Key size (bytes)
128	110
256	124
320	139
384	157
512	176
576	198
640	223
704	251
768	283
832	319
896	359
960	404
1024	417

From the Table 6.11, it is seen that with increasing key size and with varying key size, the performance of model is not affected since, the encoded key size is lesser than the original key size in fuzzy filter based security.

Table 6.12: Signature Time

Key size (bits)	Signature Time (ms)
128	0.119
256	0.147
320	0.182
384	0.226
512	0.280
576	0.348
640	0.431
704	0.535
768	0.663
832	0.822
896	1.020
960	1.264
1024	1.291

From the Table 6.12, it is seen that with increasing key size, the performance of model is not affected since the signature time is increasing at a marginal rate and not at linear rates in fuzzy filter based security.

Table 6.13: Key Generation Time (ms)

Key size (bits)	Minimum	Average	Maximum
128	9449.523	9568.305	9689.050
256	9719.482	9847.098	9950.173
320	10926.932	11036.878	11127.192
384	11061.420	11162.532	11261.680
512	11358.865	11517.895	11626.860
576	12516.250	12612.453	12746.942
640	12889.283	13039.478	13129.792
704	13245.628	13386.988	13510.678
768	13222.068	13370.300	13561.725
832	14994.958	15124.538	15218.778
896	15214.852	15354.248	15483.828

960	15896.128	15964.845	16137.618
1024	16681.912	16789.167	16913.971

From the Table 6.13, it is seen that with increasing key size, the performance of model is not affected since the key generation time in terms of minimum time, average time and maximum time is increasing at a marginal rate and not at linear rates in fuzzy filter based security. It is seen that that the average time is the iterations of all rounds between the minimum time and the maximum time.

Table 6.14: Network Performance Metrics

Metrics	Key size (bits)	
	128 (minimum)	**1024 (maximum)**
Key size (bits)	119.15s	160.25s
Delay Time	536 ms	589 ms
Average Throughput	81.25 hits/second	82.99 hits/second
Average Response time	5.21 ms	50.74 ms
Error Rate	0.1 %	0.1 %

The Table 6.14 shows reduced error rate in least percentages, reduced delay in milliseconds and there is a marginal difference between the minimum bits and maximum bits. Also, the average throughput is not a variable one and it maintains at constant rate for the minimum and maximum bit.

Table 6.15: Load vs. Cost

Load (VMs)	**Cost (USD)**
5	2.46
10	3.07
15	3.83
20	4.78
25	5.97
30	7.45
35	9.30
40	11.60

45	14.48
50	14.96

The Table 6.15 shows cost of VMs for resource allocation and storage for allocating higher rate of data. It is seen that allocation of increasing VMs tends to increase the cost at marginal rate.

6.5. SUMMARY

The results of simulation on serverless network metrics shows that the filter based security using fuzzy logic achieves higher average throughput and reduced delay, average response, error rate, load distribution than attribute based encryption and identity based cryptosystem. Further, the testing on security metrics shows that the filter based security using fuzzy obtains reduced encoded key size, signature time and key generation time with reduced minimum, average and maximum time than attribute based encryption and identity based cryptosystem. Further, the study shows that the filter based security using fuzzy is cost-efficient than attribute based encryption and identity based cryptosystem.

CHAPTER -VII

CONCLUSION AND FUTURE WORK

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1. CONCLUSION

Serverless computing is a new paradigm in cloud computing, where the cloud service provider manages the resource allocation and pricing is based on actual usage. serverless computing started moving into the focus of the industry as the concept is promising and major public cloud service providers have been pushing their runtime models to the market.

The event-driven nature of serverless computing using stateless cloud functions enables instant scaling of cloud applications. The resource management lies in the responsibility of the cloud service provider, so that the consumer does not have to worry about the infrastructure anymore. Moreover, resources are billed based on the actual usage compared to provisioned resources irrespectively of their utilization in the traditional cloud computing billing model of e.g. VMs.

Therefore, serverless computing is seen as a chance for substantial cost reduction of cloud applications. Although the cost model is more complex, besides low usage applications, especially bursty and compute-intensive application benefit from the serverless computing.

A subset of wireless computing is the serverless environment, where there are no access points and a network is defined simply by other nearby nodes. The focus of this thesis is the ad-hoc serverless network topography. This environment is best described through example. Suppose a group of corporate officials meet on a job site to discuss and plan future developments. As many job sites are remote, it must be assumed there are no servers to provide security services such as key generation or authentication.

For security reasons, it is important to be sure all meaningful communications are encrypted, and each message is verified as both unmodified and sent from whom it reports its sender to be. Each of these small devices are battery powered. The ability to form a logical grouping of these devices, generate a shared key using influences from

each member, and communicate through these devices securely is needed.

In the first part of the study, an identity based cryptosystem for secured authentication is designed that uses ECC model for securing the data in serverless environment. The segregation of initialization phase and authentication phase enables the ECC model to improve the security and authentication of data in serverless environment.

In the second part of the study, an attribute based encryption is designed under different security requirements that includes: data confidentiality, collision resistance, attack resistance, non-accessibility of sensitive before release time and delete data after expiration time. Considering all these parameters, the encryption model offers improved security of data that gets traversed or stored in the serverless environment.

In the third part of the study, a filter based security approach is developed that undergoes three different mechanism including encrypted fuzzy based filter with link reliability estimation, VM reliability estimation and residual energy factor estimation. Secondly it includes the adoption of signature verification scheme involving encryption and decryption mechanism.

Thus an experimental analysis is conducted between all the three proposed models say 1) Identity based cryptosystem 2) Attribute based encryption 3) Filter based security. The experimental analysis is conducted in terms of different security metrics that includes: encoded key size, signature time and key generation time in terms of its minimum, average and maximum time. Further, it is tested in terms of serverless network metrics that includes delay time, average throughput, average response time, error rate, load distribution and cost-efficiency.

The results of simulation on serverless network metrics shows that the filter based security using fuzzy logic achieves higher average throughput and reduced delay, average response, error rate, load distribution than attribute based encryption and identity based cryptosystem. Further, the testing on security metrics shows that the filter based security using fuzzy obtains reduced encoded key size, signature time and key generation time with reduced minimum, average and maximum time than attribute based encryption and identity based cryptosystem. Further, the study shows that the filter based security using fuzzy is cost-efficient than attribute based encryption and identity based cryptosystem.

7.2. LIMITATIONS AND FUTURE WORK

Serverless computing is getting more attention and picking up speed. Therefore, it become inevitable to examine the field in an academic setting. Research exists but is still limited due to how recent the developments are.

- Firstly, the study is not carried out in complete stages in the design science process because it does not justify and communicate the results with society. This constraint can be overcome by observational analysis of commercial programs that take account of serverless architecture and experience. Researchers track and gather information for the study of the method followed.

- Secondly, serverless prototype application creation is restricted to a specific context. As a result, the results of this study are likely to be counterfeit when introducing them in the industry with other cloud providers.

It is clear that in a serverless environment, reliance upon a database of any form of usernames and passwords is prohibited. This database would need to be replicated to every node and painfully kept current. It is very clear that password-based authentication simply will not suffice in this paradigm.

Digital certificates, on the other hand, would provide a more robust and scalable system for user authentication. A user would generate a certificate, and have the appropriate third parties (such as departments, organizations, etc.) sign this certificate, building a hierarchy and web of authentication as needed. When joining a group, the user could present this certificate, and the other nodes could quickly and safely ascertain the validity of the users credentials. The third party signatures that were gathered would serve to provide an access control list feature; if a particular network group is established requiring certain credentials and a certificate is presented without that the proper signatures, the certificate is rejected and the user is unable to join. This form of access control and user authentication is similar to techniques used in popular web server software.

Of course, this is not without technical difficulties as well. These certificates would need to be relatively short-lived to maintain correct access control. Resource changes (i.e. people moving between departments or projects) would need to be reflected in these certificate signing chains. Revocation lists and proper signing sequences would

need to be maintained, and each device would need to be synchronized with these changes. While somewhat better than the user-password database, there are many commonalities.

A goal of computing in general is improving performance across the board. We can improve computation by using efficient algorithms, but network communication and utilization provides a second area of opportunity for performance enhancement.

Analysis of the keys that are transported during the protocol phases reveals an interesting optimization opportunity. Each key transports a great deal of the key generation parameter information. While this might be important for blind key agreement techniques where each party does not know the parameters, this environment requires parameters be public. Key material could be separated from the parameter data and transported for a simple optimization. This would require object construction on the receiver side for each key, but this is, in a sense, already occurring.

Additionally, an excellent way to improve network performance is a decrease in network traffic. If a group is willing to take the additional computational impact, all messages could easily be compressed using a high-speed compression routine. Simply reducing the data on the network will improve overall network performance.

CHAPTER -VIII

REFERENCES

REFERENCES

[1] Kalapatapu, A and Sarkar, M 2012, “Cloud computing: An overview”, Cloud Computing: Methodology, Systems and Applications, pp. 1-28.

[2] Dai Y, Wu B, Gu Y, Zhang Q, & Tang C, ‘Data Security Model for Cloud Computing’, In Proceedings of the 2009 International Workshop on Information Security & Application (IWISA 2009), pp 141-144.

[3] Takabi, H, Joshi, JB and Ahn, GJ 2010, “Security and privacy challenges in cloud computing environments”, IEEE Security and Privacy, vol. 8, no. 6, pp. 24-31, ISSN: 1540-7993.

[4] Xiao, Z and Xiao, Y 2013, “Security and privacy in cloud computing”, IEEE Communications Surveys and Tutorials, vol. 15, no. 2, pp. 843- 859, ISSN: 1553-877X.

[5] Hamouda, S 2012, “Security and privacy in cloud computing”, in International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM), pp. 241-245.

[6] Ng, WK, Wen, Y and Zhu, H 2012, “Private data deduplication protocols in cloud storage”, in Proceedings of the 27th Annual ACM Symposium on Applied Computing, pp. 441-446.

[7] Luo, S and Hou, M 2013, “A novel chunk coalescing algorithm for data deduplication in cloud storage”, in IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), pp. 1-5.

[8] Leesakul, W, Townend, P and Xu, J 2014, “Dynamic data deduplication in cloud storage”, in 8th IEEE International Symposium on Service Oriented System Engineering (SOSE), pp. 320-325.

[9] Madhubala, G, Priyadharshini, R, Ranjitham, P and Baskaran, S 2014, “Nature-Inspired enhanced data deduplication for efficient cloud storage”, in International Conference on Recent Trends in Information Technology (ICRTIT), pp. 1-6.

[10] Miao, M, Jiang, T and You, I 2015, “Payment-based incentive mechanism for secure cloud deduplication”, International Journal of Information Management, vol. 35, no. 3, pp. 379-386, ISSN: 0268- 4012.

[11] Miao, M, Wang, J, Li, H and Chen, X 2015, “Secure multi-server-aided data deduplication in cloud computing”, Pervasive and Mobile Computing, vol. 24, pp. 129-137. ISSN 1574-1192.

[12] Waghmare, V and Kapse, S 2016, “Authorized Deduplication: An Approach for Secure Cloud Environment”, Procedia Computer Science, vol. 78, pp. 815-823, ISSN 1877-0509.

[13] Haoran, W, Weiqin, T, Qiang, G and Shengan, Z 2015, “A data deduplication method in the cloud storage based on FP-tree”, in 4th IEEE International Conference on Computer Science and Network Technology

(ICCSNT), vol. 1, pp. 557-562.

[14] Wu, S, Li, K. C, Mao, B and Liao, M 2017, “Dac: Improving storage availability with deduplication-assisted cloud-of-clouds”, Future Generation Computer Systems, vol. 74, pp. 190-198, ISSN: 0167- 739X.

[15] Wen, Z, Luo, J, Chen, H, Meng, J, Li, X and Li, J 2014, “A verifiable data deduplication scheme in cloud computing”, in International IEEE Conference on Intelligent Networking and Collaborative Systems (INCoS), pp. 85-90.

[16] Puzio, P, Molva, R, Onen, M and Loureiro, S 2013, “ClouDedup: secure deduplication with encrypted data for cloud storage”, in IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), vol. 1, pp. 363-370.

[17] Jin, X, Wei, L, Yu, M, Yu, N and Sun, J 2013, “Anonymous deduplication of encrypted data with proof of ownership in cloud storage”, in IEEE/CIC International Conference on Communications in China (ICCC), pp. 224-229.

[18] Rashid, F, Miri, A and Woungang, I 2012, “A secure data deduplication framework for cloud environments”, in IEEE Tenth Annual International Conference on Privacy, Security and Trust (PST), pp. 81-87.

[19] Fan, CI, Huang, SY, and Hsu, WC 2015, “Encrypted Data Deduplication in Cloud Storage”, in 10th Asia Joint Conference on Information Security (AsiaJCIS), pp. 18-25.

[20] Widodo, RN, Lim, H and Atiquzzaman, M 2017, “A new content- defined chunking algorithm for data deduplication in cloud storage”, Future Generation Computer Systems, vol. 71, pp. 145-156, ISSN: 0167-739X.

[21] Fu, Y, Jiang, H, Xiao, N, Tian, L, Liu, F and Xu, L 2014, “Application-aware local-global source deduplication for cloud backup services of personal storage”, IEEE transactions on parallel and distributed systems, vol. 25, no. 5, pp. 1155-1165, ISSN: 1045-9219.

[22] Li, J, Li, YK, Chen, X, Lee, PP and Lou, W 2015, “A hybrid cloud approach for secure authorized deduplication”, IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 5, pp. 1206-1216, ISSN:1045-9219.

[23] Gan, Q, Wang, X and Wu, D 2017, “Revocable Key-Aggregate Cryptosystem for Data Sharing in Cloud”, Security and Communication Networks, vol. 2017, pp. 1-11, ISSN:1939-0114.

[24] Kim, J and Nepal, S 2016, “A cryptographically enforced access control with a flexible user revocation on untrusted cloud storage”, Data Science and Engineering, vol. 1, no. 3, pp. 149-160, ISSN: 2364- 1185.

[25] Imine, Y, Lounis, A and Bouabdallah, A 2017, “ABR: A new efficient attribute based revocation on access control system”, in 13th International conference on Wireless Communications and Mobile Computing Conference (IWCMC), pp. 735-740.

[26] Xu, Z and Martin, KM 2012, “Dynamic user revocation and key refreshing for attribute-based encryption in cloud storage”, in 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 844-849.

[27] Zhang, P, Chen, Z, Liang, K, Wang, S and Wang, T 2016, “A cloud- based access control scheme with user revocation and attribute update”, in Australasian Conference on Information Security and Privacy, pp. 525-540.

[28] Mohan, L and Elayidom, MS 2015, “Fine Grained Access Control and Revocation for Secure Cloud Environment–A Polynomial Based Approach”, Procedia Computer Science, vol. 46, pp. 719-724.

[29] Devi, KJ and Kanimozhi, S 2014, “Efficient user revocation for dynamic groups in the cloud”, International Journal of Engineering and Computer Science, vol. 3, no. 2, pp. 3938-3942, ISSN : 2319-7242.

[30] Balani, N and Ruj, S 2014, “ Temporal access control with user revocation for cloud data”, in 13th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 336-343.

[31] Deshmukh, AR, Mante, RV and Chatur, PN 2017, “Cloud Based Deduplication and Self Data Destruction”, in IEEE International Conference on Recent Trends in Electrical, Electronics and Computing Technologies (ICRTEECT), pp. 155-158.

[32] Zeng, L, Shi, Z, Xu, S and Feng, D 2010, “Safevanish: An improved data self-destruction for protecting data privacy”, in IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), pp. 521-528.

[33] Zeng, L, Wang, Y and Feng, D 2015, “CloudSky: a controllable data self-destruction system for untrusted cloud storage networks”, In 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 352-361.

[34] Xiong, J, Yao, Z, Ma, J, Liu, X and Li, Q 2013, “A secure document self-destruction scheme: an ABE approach”, in High Performance Computing and Communications and 10th IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), pp. 59-64.

[35] Guechi, FA and Maamri, R 2017, “Secure Self-Destruction of Shared Data in Multi-CloudIoT”, in 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 161-168.

[36] Kamplee, M and Solunke, B 2017, “Data sharing and self-destruction scheme in cloud”, International Journal of Computer Applications, vol. 178, no. 4, pp. 41-45, ISSN: 0952-8091.

[37] Xiong, J, Liu, X, Yao, Z, Ma, J, Li, Q, Geng, K and Chen, PS 2014, “A secure data self-destructing scheme in cloud computing” IEEE Transactions on Cloud Computing, vol. 2, no. 4, pp. 448-458, ISSN: 2168-7161.

[38] Gadhve, S and Naidu, D 2016, “Self destruction model for protecting data in cloud storage based on data storage center”, International Journal of Engineering and Innovative Technology, vol. 6, pp. 3, pp. 32-37, ISSN 2277-3754.

[39] Yang, T, Li, J and Yu, B 2015, “A secure ciphertext self-destruction scheme with attribute-based encryption”, Mathematical Problems in Engineering, vol. 2015, pp. 1-8, ISSN: 1563-5147.

[40] Lalitha, K and Devi, SJ 2014, “SEDAS: A self destruction for protecting data privacy in cloud storage as a service model”, International Journal of Innovative Research in Science, Engineering and Technology, vol. 3, no. 1, pp. 401-404, ISSN: 2319-8753.

[41] Wang, J and Le, J 2010, “Based on private matching and min-attribute generalization for privacy preserving in cloud computing”, in Sixth IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), pp. 735-738.

[42] Lee, S and Choi, D 2012, “Privacy-preserving cross-user source-based data deduplication in cloud storage”, in International Conference on ICT Convergence (ICTC), pp. 329-330.

[43] Yang, X, Lu, R, Choo, KKR, Yin, F and Tang, X 2017, “Achieving Efficient and Privacy-Preserving Cross-Domain Big Data Deduplication in Cloud”, IEEE Transactions on Big Data, pp. 1-12, ISSN : 2332-7790.

[44] Ibrahim, A, Jin, H, Yassin, AA, Zou, D and Xu, P 2014, “Towards efficient yet privacy-preserving approximate search in cloud computing”, The Computer Journal, vol. 57, no. 2, pp. 241-254, ISSN · 0010-4620.

[45] Pan, Y, Xiaolin, G, Jian, A, Jing, Y, Jiancai, L and Feng, T 2014, “A retrievable data perturbation method used in privacy-preserving in cloud computing”, China Communications, vol. 11, no. 8, pp. 73-84, ISSN 1673-5447.

[46] Sharifi, L and Beisafar, MH 2013, “User-side personalization considering privacy preserving in cloud systems”, in 27th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 797-802.

[47] Wang, C, Chow, SS, Wang, Q, Ren, K and Lou, W 2013, “Privacy-preserving public auditing for secure cloud storage”, IEEE transactions on computers, vol. 62, no. 2, pp. 362-375, ISSN: 0018-9340.

[48] Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C. C., Khandelwal, A., Pu, Q., ... & Gonzalez, J. E. (2019). Cloud programming simplified: A berkeley view on serverless computing. *arXiv preprint arXiv:1902.03383*.

[49] Chandramohan, D, Vengattaraman, T, Rajaguru, D and Dhavachelvan, P 2016, “A new privacy preserving technique for cloud service user endorsement using multi-agents”, Journal of King Saud University-Computer and Information Sciences, vol. 28, no. 1, pp. 37-54, ISSN: 1319-1578.

[50] Li, L, Lu, R, Choo, KKR, Datta, A and Shao, J 2016, "Privacy- preserving-outsourced association rule mining on vertically partitioned databases," IEEE Transactions on Information Forensics and Security, vol. 11, no. 8, pp. 1847-1861, ISSN: 1556-6021.

[51] He, D, Kumar, N, Wang, H, Wang, L and Choo, KKR 2017, "Privacy-preserving certificateless provable data possession scheme for big data storage on cloud", Applied Mathematics and Computation, vol. 314, pp. 31-43, ISSN: 0096-3003.

[52] Filho, G, &Barreto, P. D, 'Demonstrating data possession &uncheatable data transfer', Tech. Rep', Citeseer, (2006).

[53] Deswarte, Y, Quisquater, J, Saidane, A. 'Remote Integrity checking.' In Proceedings of the Sixth Working Conference on Integrity & Internal Control in Information Systems (IICIS). Springer, Netherlands, 2004 pp 1-11.

[54] Sebe, F, Domingo-Ferrer, J, Martinez-Balleste, A, Deswarte, Y, &Quisquater, J.J.'Efficient remote data possession checking in critical information infrastructures', IEEE Trans. Knowl. Data Eng. Vol.20, No. 8, 2008, pp. 1034–1038.

[55] Kan Y, &Xiaohua J, 'Data storage auditing service in cloud computing:challenges, methods & opportunities', WorldWide Web, spinger, DOI 10.1007/s11280-011-0138-0, 2011.

[56] Golle, P, Jarecki, S, &Mironov, I, 'Cryptographic Primitives Enforcing Communication & Storage Complexity', In Proceedings of Financial Crypto, 2002.

[57] Opera, A, Reiter, M, & Yang, K, 'Space-efficient block storage Integrity', In Proceedings of the NDSS Symposium, Citeseer, 2005.

[58] Alexander, H, Bernardo, P, Charalampos, P, & Roberto, T, 'Efficient Integrity Checking of Untrusted Network Storage', In Proceedings Of StorageSS'08, Fairfax, Virginia, October 31, 2008.

[59] Aaram, Y, Chunhui, S, &Yongdae, K, 'On Protecting Integrity & Confidentiality of Cryptographic File System for Outsourced Storage', In proc.of CCSW'09, Chicago, Illinois, USA November 13, 2009.

[60] Eu-Jin, G, Hovav, S, Nagendra, M, & Dan, B, 'SiRiUS: Securing Remote Untrusted Storage', In proceedings of the Internet Society (ISOC) Network & Distributed Systems Security (NDSS) Symposium 2003, pp. 131-145.

[61] Erel, G, &Avishai, W, 'CRUST: Cryptographic Remote Untrusted Storage without Public Keys', International Journal of Information Security, Volume 8 Issue 5, September 2009.

[62] Kallahalla,M, Riedel, E, Swaminathan, R,Wang, Q, & Fu, K. ' Plutus: scalable secure file sharing on untrusted storage', In Proceedings of the 2nd USENIX Conference on File & Storage Technologies, USENIX Association, Berkeley, CA, USA, 2003, pp. 29–42.

[63] Oualha, N, & ¨Onen, M, Roudier, Y, 'A Security Protocol for Self-Organizing Data Storage.Tech. Rep. EURECOM+2399, InstitutEurecom, France ,2008.

[64] Barsoum, A. F, &Hasan, M. A, 'Provable possession & replication of data over cloud servers', Centre For Applied Cryptographic Research (CACR), University of Waterloo, Report 2010/32, 2010, http://www.cacr.math.uwaterloo.ca/techreports/2010/cacr2010- 32.pdf.

[65] Boneh, D, Lynn, B, &Shacham, H, 'Short signatures from the weilpairing', J. Cryptol. 17, 2004, pp. 297–319.

[66] Curtmola, R, Khan, O, Burns, R, &Ateniese, G. 'MR-PDP: multiple- replica provable data possession', In Proceedings of the 2008 the 28th International Conference on Distributed Computing Systems, ICDCS '08, IEEE Computer Society, Washington, DC, USA ,2008, pp. 411– 420.

[67] Hao, Z, Zhong, S, & Yu, N, 'A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics & Public Verifiability' , IEEE Trans. Knowledge & Data Engineering, Vol. 23, No.9, 2011, pp. 1432-1437.

[68] Ateniese, G, Burns, R, Curtmola, R, Herring, J, Kissner, L, Peterson, Z, & Song, D,' Provable data possession at untrusted stores', ACM Conference on Computer & Communications Security, ACM, New York, NY, USA CCS,2007,pp. 598–609.

[69] Ateniese, G, Kamara, S, & Katz, J, 'Proofs of storage from homomorphic identification protocols', In Proceedings of the 15th International Conference on the Theory & Application of Cryptology & Information Security:Advances in Cryptology,ASIACRYPT '09, Springer, Berlin, Heidelberg,2009, pp. 319–333.

[70] Blum, M, Evans,W, Gemmell, P, Kannan, S, &Naor, M, 'Checking the correctness of memories', In Proceedings of the 32nd Annual Symposium on Foundations of Computer Science, SFCS '91 IEEE Computer Society, Washington, DC, USA,1991, pp. 90–99.

[71] Naor, M, &Rothblum, G.N, 'The complexity of online memory checking', In Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS'05,. IEEE Computer Society, Washington, DC, USA, 2005, pp. 573–584.

[72] Ateniese, G, Di Pietro, R, Mancini, L.V, &Tsudik, G, 'Scalable & efficient provable data possession', In Proceedings of the 4th International Conference on Security & Privacy in Communication Networks, SecureComm '09, ACM, New York, NY, USA,2008, pp. 1- 10.

[73] Chow, R, Golle, P, Markus, J, Elaine S, Jessica, S, Ryusuke, M, & Molina, J, Chun, B.-G, Dabek, F, Haeberlen, A, Sit, E, Weatherspoon, H, Kaashoek, M. F, Kubiatowicz, J,& Morris, R, 'Efficient replica maintenance for distributed storage systems,' In NSDI'06: Proceedings of the 3rd Conference on Networked Systems Design & Implementation, Berkeley, CA, USA, 2006

[74] Curtmola, R, Khan, O, & Burns, R, 'Robust remote data checking' In Proceedings of the 4th ACM International Workshop on Storage Security & Survivability, StorageSS '08 ACM, New York, NY, USA ,2008 , pp. 63–68.

[75] Hao Z, & Yu, N, 'A multiple-replica remote data possession checking protocol with public verifiability,' In proceedings of Second International Symposium on Data, Privacy, & E-Commerce , 2010.

[76] Barsoum, A. F, &Hasan, M. A, 'On Verifying Dynamic Multiple Data Copies over Cloud Servers', Technical Report, Department of Electrical & Computer Engineering University of Waterloo, Ontario, Canada, Aug 2011.

[77] Luo, W, &Bai, G, 'Multi-Copy Privacy-Preserving Verification for Cloud Computing' International Journal of Advancements in Computing Technology(IJACT)', Vol.3, No. 9, October, 2011, pp. 9- 16.

[78] Lillibridge, M, Elnikety, S, Birrell, A, Burrows, M, &Isard, M. 'A cooperative internet backup scheme', In Proceedings of the Annual Conference on USENIX Annual Technical Conference, USENIX Association, Berkeley, CA, USA, 2003.

[79] Rabin M, 'Efficient Dispersal of Information for Security, LoadBalancing, & Fault Tolerance', J. ACM, Vol. 36, No. 2, 1989.

[80] Plank, J.S. 'A tutorial on reed-Solomon coding for fault-tolerance in raid-like systems', Technical Report UT-CS-96-332, University of Tennessee, July, 1996.

[81] Schwarz, T, & Miller, E. 'Store, forget, & check: Using algebraic signatures to check remotely administered storage.' In Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06), 2006.

[82] Carter, L, Wegman, M, 'Universal Hash Functions (UHF),' Journal of Computer & System Sciences, Vol. 18, No. 2, 1979, pp. 143–154.

[83] Juels, A, &Kaliski, Jr, B.S. 'POR: proofs of retrievability for large files', In Proceedings of the 14th ACM Conference on Computer & Communications Security, CCS '07, ACM, New York, NY, USA,2007, pp. 584–597.

[84] Shacham, H, & Waters, B. 'Compact proofs of retrievability', In Proceedings of the 14th International Conference on the Theory & Application of Cryptology & Information Security: Advances in Cryptology, ASIACRYPT '08, Springer, Berlin, Heidelberg, 2008, pp. 90–107.

[85] Bowers, K.D, Juels, A, &Oprea, A, 'HAIL:A High-Availability & Integrity Layer for Cloud Storage', In CCS '09 Proceedings of the 16th ACM conference on Computer & communications security, ACM New York, NY, USA,2009.

[86] Dodis, Y, Vadhan, S, Wichs, D. 'Proofs of retrievability via hardness amplification', In Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography, TCC '09, Springer, 2009, pp. 109–127.

[87] Bowers, K.D, Juels, A, &Oprea, A, 'Proofs of Retrievability: theory & implementation', In Proceedings of the 2009 ACM Workshop on Cloud Computing Security, CCSW '09, ACM ,New York, NY, USA,2009 , pp. 43–54.

[88] Ateniese, G, Burns, R, Curtmola, R, Herring, J, Khan, O, Kissner, L, Peterson, Z, & Song, D, ' Remote Data Checking Using Provable Data Possession', ACM Transactions on Information & System Security, Vol. 14, No. 1 , Article 12, May 2011, pp. 12.1-12.34.

[89] Chen, L, &Guo, G, 'An Efficient Remote Data Possession Checking in Cloud Storage', International Journal of Digital Content Technology & its Applications. Vol. 5, No. 4, April 2011, pp. 43-50.

[90] Rivest, R. Shamir, A.&Adleman, L. 'A method for obtaining digital signatures & public-key cryptosystems,'Commun. ACM, Vol. 26, No. 1, 1983.

[91] Krawczyk, H, 'Distributed Fingerprints & Secure Information Dispersal,' In Proceedings 12th Ann. ACM Symp. Principles of Distributed Computing (PODC) ,1993.

[92] Hou, F, He, H, Xiao, N, Liu, F, Zhong, G, 'Static, Dynamic &Incremental MAC Combined Approach for Storage Integrity Protection', 2010 10th IEEE International Conference on Computer & Information Technology (CIT 2010).

[93] Li, M, Yu, S, Ren, K, Lou, W. 'Securing personal health records in cloud computing: patient-centric & fine-grained data access control in multi-owner settings', Security & Privacy in Communication Networks', 2010, pp. 89–106.

[94] Jia Yu, Fanyu Kong, &RongHao, 'Publicly Verifiable Secret Sharing with Enrollment Ability', Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, & Parallel/Distributed Computing.

[95] Wang, B, Li, B and Li, H 2014, "Oruta: Privacy-preserving public auditing for shared data in the cloud", IEEE transactions on cloud computing, vol. 2, no. 1, pp. 43-56, ISSN: 2168-7161.

[96] Plank, J. S, 'Jerasure: A library in C/C++ facilitating erasure coding for storage applications Tech', Rep. CS-07-603, University of Tennessee, September, 2007.

[97] Plank. J. S, 'Optimizing Cauchy Reed-Solomon codes for fault-tolerant storage applications', Technical Report CS-05-569, Univ. Tennessee, December 2005.

[98] Gopalan, S, Charles, P. W, &Erez, Z, 'Ensuring Data Integrity in Storage: Techniques & Applications', In Proceedings of StorageSS'05, Fairfax, Virginia, USA, November 11, 2005.

[99] Wang, B, Li, B and Li, H 2012, "Oruta: Privacy-preserving public auditing for shared data in the cloud", in 5th IEEE International Conference on Cloud Computing (CLOUD), pp. 295-302.

[100] Ren, W, Ren , Y, &Zhag , H, 'Secure, dependable & publicly verifiable distributed data storage in unattended wireless sensor networks', Sci. China InfSci , Vol. 53: No. 5, May, 2010, pp. 964–979.

[101] C. Chris Erway,Alptekin Küpçü,Charalampos Papamanthou, Roberto Tamassia, "Dynamic Provable Data Possession", in ACM Transactions on Information and System Security ,2009,17(4):213-222

[102] Gruschka, N, &Iacono, L. L, 'Vulnerable Cloud: SOAP Meaage Security Validation Revisited', In Procedings of IEEE International Conference on Web Service, Jul 2009, pp. 635-631.

[103] Wang, C, Wang, Q, Ren, K and Lou, W 2010, "Privacy-preserving public auditing for data storage security in cloud computing", in Infocom 2010 proceedings IEEE, pp. 1-9.

[104] Merkle, R.C. 'Protocols for public key cryptosystems', In proceedings of IEEE Symposium on Security & Privacy, 1980, pp. 122-134.

[105] Wu, Y, Jiang, ZL, Wang, X, Yiu, SM and Zhang, P 2017, "Dynamic Data Operations with Deduplication in Privacy-Preserving Public Auditing for Secure Cloud Storage", in IEEE International Conference on Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), vol. 1, pp. 562-567.

[106] Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., ... & Suter, P. (2017). Serverless computing: Current trends and open problems. In *Research Advances in Cloud Computing* (pp. 1-20). Springer, Singapore.

[107] Castro, P., Ishakian, V., Muthusamy, V., & Slominski, A. (2019). The server is dead, long live the server: Rise of Serverless Computing, Overview of Current State and Future Trends in Research and Industry. *arXiv preprint arXiv:1906.02888*.

[108] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50-58.

[109] Soltani, B., Ghenai, A., & Zeghib, N. (2018). Towards Distributed Containerized Serverless Architecture in Multi Cloud Environment. *Procedia computer science*, *134*, 121-128.

[110] Chou, T. S. (2013). Security threats on cloud computing vulnerabilities. *International Journal of Computer Science & Information Technology*, *5*(3), 79.

[111] Boneh, D., & Waters, B. (2007, February). Conjunctive, subset, and range queries on encrypted data. In *Theory of Cryptography Conference* (pp. 535-554). Springer, Berlin, Heidelberg.

[112] Kakkad, V., Patel, M., & Shah, M. (2019). Biometric authentication and image encryption for image security in cloud framework. *Multiscale and Multidisciplinary Modeling, Experiments and Design*, *2*(4), 233-248.

[113] Koo, D., Hur, J., & Yoon, H. (2013). Secure and efficient data retrieval over encrypted data using attribute-based encryption in cloud storage. *Computers & Electrical Engineering*, *39*(1), 34-46.

[114] Fehr, S., & Salvail, L. (2017, April). Quantum authentication and encryption with key recycling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 311-338). Springer, Cham.

[115] Luan, G., Li, A., Zhang, D., & Wang, D. (2018). Asymmetric image encryption and authentication based on equal modulus decomposition in the Fresnel transform domain. *IEEE Photonics Journal*, *11*(1), 1-7.

[116] Liu, T., Tian, J., Gui, Y., Liu, Y., & Liu, P. (2017). SEDEA: State estimation-based dynamic encryption and authentication in smart grid. *IEEE Access*, *5*, 15682-15693.

[117] Unterluggauer, T., Werner, M., & Mangard, S. (2019). MEAS: memory encryption and authentication secure against side-channel attacks. *Journal of cryptographic engineering*, *9*(2), 137-158.

[118] Chen, J., Bao, N., Zhang, L. Y., & Zhu, Z. L. (2018). Optical information authentication using optical encryption and sparsity constraint. *Optics and Lasers in Engineering*, *107*, 352-363.

[119] Kwon, H., Kim, D., Hahn, C., & Hur, J. (2017). Secure authentication using ciphertext policy attribute-based encryption in mobile multi-hop networks. *Multimedia Tools and Applications*, *76*(19), 19507-19521.

[120] Yu, R., Wang, J., Xu, T., Gao, J., An, Y., Zhang, G., & Yu, M. (2017). Authentication with block-chain algorithm and text encryption protocol in calculation of social network. *IEEE Access*, *5*, 24944-24951.

[121] Yuan, L., Ran, Q., & Zhao, T. (2017). Image authentication based on double-image encryption and partial phase decryption in nonseparable fractional Fourier domain. *Optics & Laser Technology*, *88*, 111-120.

[122] Namasudra, S., & Roy, P. (2017). A new secure authentication scheme for cloud computing environment. *Concurrency and Computation: Practice and Experience*, *29*(20), e3864.

[123] Namasudra, S., & Roy, P. (2016). Secure and efficient data access control in cloud computing environment: A survey. *Multiagent and Grid Systems*, *12*(2), 69-90.

[124] Zhuge, H., & Sun, X. (2008). A virtual ring method for building small-world structured P2P overlays. *IEEE Transactions on Knowledge and Data Engineering*, *20*(12), 1712-1725.

[125] Namasudra, S., Roy, P., Balusamy, B., & Vijayakumar, P. (2017, March). Data accessing based on the popularity value for cloud computing. In *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)* (pp. 1-6). IEEE.

[126] Zhuge, H., Xing, Y., & Shi, P. (2008). Resource space model, OWL and database: Mapping and integration. *ACM Transactions on Internet Technology (TOIT)*, *8*(4), 1-31.

[127] Devi, D., & Purkayastha, B. (2017). Redundancy-driven modified Tomek-link based undersampling: a solution to class imbalance. *Pattern Recognition Letters*, *93*, 3-12.

[128] Namasudra, S., & Roy, P. (2018). PpBAC: popularity based access control model for cloud computing. *Journal of Organizational and End User Computing (JOEUC)*, *30*(4), 14-31.

[129] Douceur, J. R., & Wattenhofer, R. P. (2001, October). Optimizing file availability in a secure serverless distributed file system. In *Proceedings 20th IEEE Symposium on Reliable Distributed Systems* (pp. 4-13). IEEE.

[130] King, M., & Muehlemann, M. (2002, September). Transforming the reliability, security and scalability of IT communications through the pervasive deployment of serverless software infrastructure. In *Proceedings. Second International Conference on Peer-to-Peer Computing,* (pp. 5-12). IEEE.

[131] Ahamed, S. I., Rahman, F., Hoque, E., Kawsar, F., & Nakajima, T. (2008, April). S3PR: Secure serverless search protocols for RFID. In *2008 International Conference on Information Security and Assurance (isa 2008)* (pp. 187-192). IEEE.

[132] Shin, Y., Koo, D., Yun, J., & Hur, J. (2016, December). SEED: Enabling Serverless and Efficient Encrypted Deduplication for Cloud Storage. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 482-487). IEEE.

[133] Bila, N., Dettori, P., Kanso, A., Watanabe, Y., & Youssef, A. (2017, June). Leveraging the serverless architecture for securing linux containers. In *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)* (pp. 401-404). IEEE.

[134] Parres-Peredo, A., Piza-Davila, I., & Cervantes, F. (2019, July). Building and Evaluating User Network Profiles for Cybersecurity Using Serverless Architecture. In *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)* (pp. 164-167). IEEE.

[135] McGrath, G., & Brenner, P. R. (2017, June). Serverless computing: Design, implementation, and performance. In *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)* (pp. 405-410). IEEE.

[136] Buyya, R., Srirama, S. N., Casale, G., Calheiros, R., Simmhan, Y., Varghese, B., ... & Toosi, A. N. (2018). A manifesto for future generation cloud computing: Research directions for the next decade. *ACM computing surveys (CSUR), 51*(5), 1-38.

[137] Dean, J., Harrison, A., Lass, R. N., Macker, J., Millar, D., & Taylor, I. (2011). Client/server messaging protocols in serverless environments. *Journal of network and computer applications, 34*(4), 1366-1379.

[138] Sultan, S., Ahmad, I., & Dimitriou, T. (2019). Container security: Issues, challenges, and the road ahead. IEEE Access, 7, 52976-52996.

[139] Qin, J., Li, H., Xiang, X., Tan, Y., Pan, W., Ma, W., & Xiong, N. N. (2019). An encrypted image retrieval method based on Harris corner optimization and LSH in cloud computing. IEEE Access, 7, 24626-24633.

[140] Agrawal, N., & Tapaswi, S. (2019). A trustworthy agent-based encrypted access control method for mobile cloud computing environment. Pervasive and Mobile Computing, 52, 13-28.

[141] Sajay, K. R., Babu, S. S., & Vijayalakshmi, Y. (2019). Enhancing the security of cloud data using hybrid encryption algorithm. Journal of Ambient Intelligence and Humanized Computing, 1-10.

[142] Veerabathiran, V. K., Mani, D., Kuppusamy, S., Subramaniam, B., Velayutham, P., Sengan, S., & Krishnamoorthy, S. (2020). Improving secured ID-based authentication for cloud computing through novel hybrid fuzzy-based homomorphic proxy re-encryption. Soft Computing, 1-16.

[143] Geng, Y. (2019). Homomorphic encryption technology for cloud computing. Procedia Computer Science, 154, 73-83.

[144] Liu, P. (2020). Public-Key Encryption Secure Against Related Randomness Attacks for Improved End-to-End Security of Cloud/Edge Computing. IEEE Access, 8, 16750-16759.

[145] Li, J., Wang, S., Li, Y., Wang, H., Wang, H., Wang, H., ... & You, Z. (2019). An efficient attribute-based encryption scheme with policy update and file update in cloud computing. IEEE Transactions on Industrial Informatics, 15(12), 6500-6509.

[146] Von Arb, M., Bader, M., Kuhn, M., & Wattenhofer, R. (2008, October). Veneta: Serverless friend-of-friend detection in mobile social networking. In 2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (pp. 184-189). IEEE.

[147] Shila, D. M., Cheng, Y., & Anjali, T. (2010). Mitigating selective forwarding attacks with a channel-aware approach in WMNs. IEEE transactions on wireless communications, 9(5), 1661-1675.

[148] Fischlin, M., & Günther, F. (2020, February). Modeling memory faults in signature and authenticated encryption schemes. In Cryptographers' Track at the RSA Conference (pp. 56-84). Springer, Cham.

[149] Zhou, Y., Li, Z., Hu, F., & Li, F. (2019). Identity-based combined public key schemes for signature, encryption, and signcryption. In Information Technology and Applied Mathematics (pp. 3-22). Springer, Singapore.

[150] Prasad, A., & Kaushik, K. (2019). Digital Signatures. Emerging Security Algorithms and Techniques, 249.

[151] Prasetyadi, G., Hantoro, U. T., Mutiara, A. B., Muslim, A., & Refianti, R. (2019, October). Heresy: A Serverless Web Application to Store Compressed and Encrypted Document in the Form of URL. In 2019 Fourth International Conference on Informatics and Computing (ICIC) (pp. 1-5). IEEE.

[152] Reisman, R. (2019). Blockchain serverless public/private key infrastructure for ADS-B security, authentication, and privacy. In AIAA Scitech 2019 Forum (p. 2203).

[153] Alexander, S, Christian, C, Asaf, C, Idit, K, Yan, M, &Dani, S, 'Venus: Verification for Untrusted Cloud Storage', In ProcedingsOf CCSW'10, Chicago, Illinois, USA October 8, 2010.

[154] Ali, M, Bilal, K, Khan, S, Veeravalli, B, Li, K and Zomaya, A 2015, "DROPS: Division and replication of data in the cloud for optimal performance and security", IEEE Transactions on Cloud computing, pp. 1-14, ISSN: 2168-7161.

[155] Anjie P, & Lei W, 'One Publicly Verifiable Secret Sharing Scheme based on Linear Code', In Proc. Of 2010 2nd Conference on Environmental Science & Information Application Technology, Jul- 2010, pp. 260-262.

[156] Arshad, J, Townend, P, &Xu, J, 'An Abstract Model for Integrated Intrusion Detection & Severity Analysis for Clouds', International Journal of Cloud Applications & Computing, 1(1), March 2011, pp. 1- 15.

[157] Ashish, K, & Elisa, B, 'Structural Signatures for Tree Data Structures', In Proceedings Of PVLDB '08, Auckland, New Zealand, August, 2008.

[158] Ashish, K, 'Tornado Codes &Luby Transform Codes', Technical Report, October, 2003.

[159] Bhaskar, P.R, Eumni, C, & Ian, L, 'Taxnomy of Cloud Computing Services', In Proceedings of Fifth International Joint Conference on INC, IMS & IDC , IEEE Computer Society Washington, DC, USA, 2009 pp 44-51 .

[160] Blomer, J, Kalfane, M, Karpinski, M, Karp, R, Luby, M, & Zuckerman, D, 'An XOR-Based Erasure-Resilient Coding Scheme', Technical Report ICSI TR-95-048, August 1995.

[161] Bonomi, F, Milito, R, Zhu, J and Addepalli, S 2012, "Fog computing and its role in the internet of things", in Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp. 13-16.

[162] Brately, P, & Fox, B. L, 'Algorithm 659: Implementing Sobol's Quasi-random Sequence Generator', ACM Trans. Math. Software 14 (1), 1988, pp. 88–100.

[163] C. Cachin, I. Keidar, & A. Shraer, Trusting the cloud, SIGACT News, vol. 40,no.2, pp. 81–86, (2009)

[164] Caronni, G, &Waldvogel, M, 'Establishing Trust in Distributed Storage Providers', In Proceedings of Third IEEE P2P Conference, Linkoping 03, 2003.

[165] Chang, E.C, &Xu, J, 'Remote Integrity check with dishonest storage server', In Proceedings of the 13th European Symposium on Research in Computer Security: Computer Security, ESORICS '08, Springer, Berlin, Heidelberg, 2008, pp. 223–237.

[166] Chi, H, 'Parallel quasirandom number generations for heterogeneous computing environments' , International Journal of Parallel, Emergent & Distributed Systems ,Vol. 24, No. 1, February 2009, pp. 21–29.

[167] Chin-Fu K, Ai-Chun, P, & Sheng-Kun, C, 'Dynamic Routing with Security Considerations', IEEE Transaction on Parallel & Distributed Systems, Vol. 20, No. 1, January 2009, pp. 48-58.

[168] Clarke, D, Devadas, S, van Dijk, M, Gassend, B, &Suh, G.E, 'Incremental multiset hash functions & their application to memory Integrity checking', In Proceedings of the 9th International Conference on the Theory & Application of Cryptology & Information Security: Advances in Cryptology, ASIACRYPT'03, pp. 188–207.

[169] Danish, J, & Hassan, Z, 'Security Issues In Cloud Computing & Computer Measures', International Journal of Engineering Science & Technology (IJEST), Vol. 3, No. 4 April 2011, pp. 2672-2676.

[170] David, M, & Dennis, S, 'Building secure file systems out of Byzantine storage', In Proceedings of the twenty-first annual symposium on Principles of distributed computing, ACM New York, NY, USA, 2002 pp 108-117.

[171] Dillon, T, Wu, C and Chang, E 2010, "Cloud computing: issues and challenges", in 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 27-33.

[172] Dwork, C, Naor, M, Rothblum, G.N, &Vaikuntanathan, V, 'How efficient can memory checking be?', In Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography,TCC '09, Springer , 2009, pp. 503–520.

[173] Fahad, A, Dr Faisal, B, &Asif, M, 'A publicly verifiable low cost signcryption scheme ensuring Confidentiality', In Prosceedings Of 2010 Second International Conference on Networks Security, Wireless Communications & Trusted Computing, 2010.

[174] Fischlin, M., & Günther, F. (2020, February). Modeling memory faults in signature and authenticated encryption schemes. In Cryptographers' Track at the RSA Conference (pp. 56-84). Springer, Cham.

[175] González-Manzano, L and Orfila, A 2015, "An efficient confidentiality-preserving proof of ownership for deduplication", Journal of Network and Computer Applications, vol. 50, pp. 49-59, ISSN:1084-8045.

[176] Greveler, U, Justus, B and Loehr, D 2011, "A privacy preserving system for cloud computing", in 11th IEEE International Conference on Computer and Information Technology (CIT), pp. 648-653.

[177] Hendricks, J, Ganger, G, & Reiter, M, 'Verifying Distributed Erasure coded Data,'Proc. 26th ACM symposium on Principles of Distributed Computing, 2007, pp. 139–146.Proceedings of CCSW'11, Chicago, Illinois, USA, October 21, 2011

[178] Hoefer, C.N, &Karagiannis, G, 'Taxonomy of cloud computing services', IEEE globecome 2010 workshop on Enabling the Future Service-Oriented Internet, 2010.

[179] Hohenberger, S, & Waters, B, 'Short & Stateless Signatures from the RSA Assumption', Advances in Cryptology CRYPTO 2009, Springer, Vol. 5677 of Lecture Notes in Computer Science, 2009, pp. 654-670.

[180] Jaeger, T, &Schiffman, J, 'Outlook: Cloud with a Chance of Security Challenges & Improvements', IEEE Trans. Security & Privacy, Article, Vol. 8, No. 1, Jan-Feb. 2010, pp. 77-80.

[181] James, B, Rajkumar, B, &Zahir T, 'MetaCDN: Harnessing Storage Clouds for High Performance Content Delivery, Journal of Network & Computer Applications, Vol.32, No. 5, Elsevier, Amsterdam, The Netherlands, Sept. 2009, pp. 1012-1022,.

[182] Jensen, M, Schwenk, J, Gruschka, N, Iacono, L.L, 'On Technical Security Issues in Cloud Computing', In ProcedingsOf 2009 IEEE International Conference on Cloud Computing. , Sep 2009, pp. 109- 116.

[183] Jesus, Ms, Bogdan, N, Gabriel, A, Alberto, S, &MarıaS, P, 'Using Global Behavior Modeling to Improve QoS in Cloud Data Storage Services', In Proceedings Of 2nd IEEE International Conference on Cloud Computing Technology & Science.

[184] John, H, Lori, M. K, & Bruce, P, 'Data Security in the World of Cloud Computing', IEEE Tranc. On Security & Privacy, Article, July-Aug, 2009.

[185] Johnson, R, et al, 'Homomorphic Signature Schemes', In Proceedings of the Cryptographer's Track at the RSA Conference on Topics in Cryptology, Springer- Verlag, 2002, pp. 244-262.

[186] Jones, G. & Jones, J. 'Elementary Number Theory,' Springer-Verlag, London, 1998.

[187] Judith, H, Robin, B, Marcia, K, & Fern, H, 'Cloud Computing FOR DUMMIES', by WILEY INDIA EDITION.

[188] Karen, S,, Wayne, J, & Miles, T, 'Guide to General Server Security', Recommendations of the National Institute of Standards & Technology, NIST Special Publication 800-123.

[189] Kevin, H, Murat, K, Lathifur, K, &Bhavani, T, 'Security Issues for Cloud Computing', Technical Report UTD-CS-02-10, Dept. of Computer Science, University of Texas, February,2010.

[190] Koyama, K, Maurer, U, Okamoto, T, & Vanstone, S, 'New Public-Key Schemes Based on Elliptic Curves over the Ring Zn', Advances in Cryptology - CRYPTO '91, Lecture Notes in Computer Science, Springer-Verlag, Vol. 576, Augest, 1991, pp. 252-266.

[191] Krawczyk, H, 'Secret Sharing Made Short,' In Proceedings of 13th Ann. Int'l Cryptology Conf. (Crypto), 1993.

[192] Lakshmanan, S, Ahamad, M, &Venkateswaran, H. 'Responsive Security for Stored Data,' IEEE Trans. Parallel & Distributed Systems, Vol. 14, No. 9, 2003.

[193] Lena W, 'Horizontal Fragmentation for Data Outsourcing with Formula-Based Confidentiality Constraints', In IWSEC'10 Proceedings of the 5th international conference on Advances in information & computer security, Springer-Verlag, Berlin, Heidelberg, 2010.

[194] Li, H, Dong, M, Liao, X and Jin, H 2014, "Deduplication-based energy efficient storage system in cloud environment", The Computer Journal, vol. 58, no. 6, pp. 1373-1383.

[195] Li, J, Krohn, M, Mazieres, D, Shasha, D. 'Secure untrusted data repository (sundr)', In Proceedings of the 6th Conference on Symposium on Operating Systems Design &Implementation,vol. 6, USENIX Association, Berkeley, CA, USA , 2004, pp. 1–9.

[196] Li, Q, Lui, J, C.S, ,& Chiu, D.M, 'On the Security & Efficiency of Content Distribution via Network Coding', IEEE Tranc. On Dependable & Secure Computing, Vol. 9, No. 2, MARCH/APRIL 2012, pp. 211-221.

[197] Li, X, Subramanyam C, & Ling L, 'Preserving Data Privacy in Outsourcing Data Aggregation Services' , ACM Transactions on Internet Technology, Vol. 7, No. 3, Article 17,August 2007, pp. 1-28.

[198] Li, Y., Yu, H., Song, B., & Chen, J. (2019). Image encryption based on a single-round dictionary and chaotic sequences in cloud computing. *Concurrency and Computation: Practice and Experience*, e5182.

[199] Lin, J-S. 'Cloud Data Storage for Group Collaborations', In Proceedings of the World Congress on Engineering 2010 Vol 1 WCE 2010, London, U.K, June - July, 2010.

[200] Liu, C, Chen, J, Yang, LT, Zhang, X, Yang, C, Ranjan, R and Kotagiri, R 2014, "Authorized public auditing of dynamic big data storage on cloud with efficient verifiable fine-grained updates", IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 9, pp. 2234-2244, ISSN:1045-9219.

[201] Liu, Q, Wang, G, & Wu, J, 'Secure & privacy preserving keyword searching for cloud storage services', Journal of Network & Computer Applications', Elsevier, Vol. 3, Issue 3, May 2012, pp. 927–933.

[202] Mao, W.B, 'Talking about the Cloud Computing', 2009-03, Available from:http://blog.csdn.net/wenbomao/archive/2009/03/03/3952761.aspx, & http://www.daoliproject.org.

[203] Martel, C. Nuckolls, G, Devanbu, P, Gertz, M, Kwong, A, &Stubblebine, S. G, 'A general model for authenticated data structures,'Algorithmica , Vol. 39, 2001.

[204] Mather, T, Kumaraswamy, S, &Latif, S, 'Cloud Security &Privacy'O'REILLY Publication, 2009.

[205] Mei, A , Mancini, L.V, &Jajodia, S, ' Secure Dynamic Fragment & Replica Allocation in Large-Scale Distributed File Systems', IEEE Trans. Parallel & Distributed Systems, Vol. 14, No. 9, 2003.

[206] Mell, P, &Grance, T, 'Effectively & Securely Using the Cloud Computing Paradigm,' National Institute Standards & Technology, October, 2009.

[207] Michael, G, Charalampos, P,, Roberto, T, & Nikos, T, 'Athos: Efficient Authentication of Outsourced File Systems', In Procedings of Information Security Conference 2008 (ISC 2008), Taipei, Taiwan, September, 2008.

[208] Miller, R, 'Amazon addresses EC2 power outages', Data Center Knowledge' , May- 2010.

[209] Miller, V, 'Uses of elliptic curves in cryptography advances in Cryptology', In Proceedings of Crypto'85, Lecture Notes in Computer Science, 218 (1986), Springer- Verlag, 1986, pp. 417-426.

[210] Minsky, Y, Trachtenberg, A, &Zippel, R, 'Set Reconciliation with Nearly Optimal Communication Complexity', IEEE Trans. Information Theory, Vol.49, No. 9, September, 2003, pp. 2213-2218.

[211] Nallakumar, MR, Sengottaiyan, N and MohamedArif, M 2014, "Cloud computing and methods for privacy preservation: a survey", International Journal of Advanced Research in Computer Engineering and Technology, vol. 3, no. 11, pp. 3752-3756, ISSN: 2278 – 1323.

[212] Nariman M, ' Report on Cloud Computing', Fall, 2008.

[213] Neela, TJ and Saravanan, N 2013, "Privacy preserving approaches in cloud: a survey", Indian Journal of Science and Technology, vol. 6, no. 5, pp. 4531-4535, ISSN: 0974-6846.

[214] Nisbet, B. 'FAS storage systems: Laying the foundation for application Availability', Network Appliance white paper: http://www.netapp. com/us/library/analyst- reports/ar1056.html, February 2008.

[215] Oded G, Shafi G, & Silvio M, 'How to Construct Random Functions', Journal of the ACM Journal of association Computing, Vol.33, No.4, 1986, pp. 792-807.

[216] Okamoto, T., Fujisaki, E., & Morita, H. (2000). PSEC: Provably secure elliptic curve encryption scheme. In IEEE P1363a.

[217] Oualha, N, &Roudier, Y, 'Probabilistically Secure Cooperative Distributed Storage', Research Report RR-07-188, February 2007, pp. 1-14.

[218] Oualha, N. ,&Roudier, Y, 'Securing adhoc storage through probabilistic cooperation assessment', WCAN '07, 3rd Workshop on Cryptography for Ad hoc Networks, Wroclaw, Poland, July 8, 2007.

[219] Patil, PV 2015, "Fog computing", in National Conference on Advancements in Alternate Energy Resources for Rural Applications, pp. 1-6.

[220] Paulo S. Barreto, L. M, &Naehrig, M, 'Ieee P1363.3 submission: Pairing-friendly elliptic curves of prime order with embedding degree 12', New Jersey: IEEE Standards Association, 2006.

[221] Pawar, PR and Waghmare, A 2015, "Data Deduplication in Cloud Storage", in National Conference on Advances in Computing, pp. 5-9.

[222] Prasad, A., & Kaushik, K. (2019). Digital Signatures. Emerging Security Algorithms and Techniques, 249.

[223] RaduSion. 'Query execution assurance for outsourced' databases. In Procedings of VLDB , ACM, 2005, pp. 601–612.

[224] Ran, C, Oded, G, &Shai, H, 'The Random Oracle Methodology Revisited', Journal of the ACM (JACM), Volume 51 Issue 4, July, 2004.

[225] Rao, KK, Hafner, L. J, & Golding, R. A, 'Reliability for Networked Storage Nodes', IEEE Trac. On Dependable & Secure Computing, Vol. 8, No. 3, MAY/JUN E 2011, pp. 404-418.

[226] Ren, Xun-Yi, Ma, Xiao-Dong, 'A* Algorithm Based Optimization for Cloud Storage', JDCTA, Vol. 4, No. 8, 2010, pp. 203 - 208.

[227] Rodrigues, R, Liskov, B, Chen, K, Liskov, M, &Schult D, 'Automatic Reconfiguration for Large-Scale Reliable Storage Systems', IEEE Tranction On Dependable & Secure Computing, Vol. 9, No. 2, March/April 2012, pp. 145-158.

[228] Sachin, A, Vikas C,,& Ari, T, 'Bandwidth Efficient String Reconciliation Using Puzzles', IEEE Tranc. ON Parallel AND Distributed Systems, Vol. 17, No. 11, Nov, 2006, pp. 1215-1225.

[229] Sadie, C, Paul, H, Siani, P, & Yun, S, ' Data Protection-Aware Design for Cloud Computing', In Procedings of CloudCom 2009, Beijing, Springer LNCS, December, 2009.

[230] Santos, N, Gummadi, K.P, & Rodrigues, R, 'Towards trusted cloud computing', In Proceedings of the 2009 conference on Hot topics in cloud computing, USENIX Association: San Diego, California, 2009.

[231] Savitha S, Thangam P 2017, "Comparative Analysis of Security Algorithms for Cloud-Based Systems and Transactions", International Journal of Control Theory and Applications (IJCTA), vol. 10, no. 39, pp. 291-299, ISSN: 0974-5572.

[232] Sen, S. K, Samanta T. & Reese, A. ‘Quasi Versus Pseudo-random Generators: Discrepancy Complexity & Integration-Error Based Comparision’, International Journal of Innovative Computing, Information & Control, Vol. 2, No. 3, 2006,pp. 621-651.

[233] Shah, M. A, Baker, M, Mogul, J.C, &Swaminathan, R, ‘Auditing to keep online storage services honest’, Proceedings of the 11th USENIX workshop on Hot Topics in Operating Systems, USENIX Association, Berkeley, CA, USA , 2007.

[234] Shah, M.A, Swaminathan, R, & Baker, M. ‘Privacy-preserving audit & extraction of digital contents’, Tech. rep, Cryptology ePrint Archive, Report 2008/186, 2008. http://eprint.iacr.org, 2008.

[235] Shila, D. M., Cheng, Y., & Anjali, T. (2010). Mitigating selective forwarding attacks with a channel-aware approach in WMNs. IEEE transactions on wireless communications, 9(5), 1661-1675.

[236] Shin, Y., Hur, J., Koo, D., & Yun, J. (2020). Toward Serverless and Efficient Encrypted Deduplication in Mobile Cloud Computing Environments. *Security and Communication Networks, 2020*.

[237] Shobana, R, Shalini, KS, Leelavathy, S and Sridevi, V 2016, “De-duplication of data in cloud”, International Journal of Chemical Sciences, vol. 14, no. 4, pp. 2933-2938, ISSN: 2523-6075.

[238] Stojmenovic, I and Wen, S 2014, “The fog computing paradigm: Scenarios and security issues”, in Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1-8.

[239] Tharunn, G, Kommineni, G, Varma, SS and Verma, AS 2015, “Data deduplication in cloud storage”, International Journal of Advanced Engineering and Global Technology, vol. 3, no. 8, pp. 1062-1065, ISSN: 2309-4893.

[240] Wang, H., Qin, Y., Huang, Y., Wang, Z., & Zhang, Y. (2017). Multiple-image encryption and authentication in interference-based scheme by aid of space multiplexing. *Optics & Laser Technology*, *95*, 63-71.

[241] Yuan, J and Yu, S 2015, “Public integrity auditing for dynamic data sharing with multiuser modification”, IEEE Transactions on Information Forensics and Security, vol. 10, no. 8, pp. 1717-1726, ISSN: 1556-6021.

[242] Zeng, L, Chen, S, Wei, Q and Feng, D 2012, “SeDas: A self- destructing data system based on active storage framework”, IEEE Transactions on Magnetics, vol. 49, no. 6, pp. 2548-2554, ISSN: 1941- 0069.

[243] Zhang, Q, Cheng, L and Boutaba, R 2010, “Cloud computing: state-of- the-art and research challenges”, Journal of internet services and applications, vol. 1, no. 1, pp. 7-18, ISSN: 1867-4828.

[244] Zhou, Y., Li, Z., Hu, F., & Li, F. (2019). Identity-based combined public key schemes for signature, encryption, and signcryption. In Information Technology and Applied Mathematics (pp. 3-22). Springer, Singapore.