**INTRODUCTION**

In the current business scenario, data leakage is a big challenge as critical organizational data should be protected from unauthorized access. Data leakage may be defined as the accidental or intentional distribution of private organizational data to the unauthorized entities. It is important to protect the critical data from being misused by any unauthorized use. Critical data include intellectual copy right information, patent information, functional information etc. In many organizations, this critical organizational data have been shared to many stakeholder outside the organizational premises. Therefore, it is difficult to identify the culprit, who has leaked the data[1][2]. In the proposed work, our goal is to identify the guilty user when the organizational data have been leaked by some agent. In the proposed work, Bell-La Padula security model has been used which provide the analysis and design of secure computer systems. This model is called data confidentiality model. Bell-LaPadula model mainly focuses on data confidentiality issues and provides controlled access to classified information. In contrast to the Biba-Integrity model which describes rule for the protection of data integrity[3]. In this formal model, the entities in an information system are divided into subjects and objects. The notion of a ”secure state” is defined, and it is proven that each state transition preserves security by moving from one secure state to other secure state, thereby inductively proving that the system satisfies the security objectives of the model. The Bell-LaPadula model is built on the concept of a state machine with a set of allowable states in a computer system. A system state is defined to be secure if the only permitted access modes of subjects to objects are in accordance with a security policy. Cloud computing has become increasingly popular due to its ability to provide on-demand computing resources and cost-effective solutions. However, with the increasing usage of cloud computing services, the risk of data breaches and data leakage has also increased. Data leakage refers to the unauthorized disclosure of confidential or sensitive information, which can lead to significant financial and reputational damage for individuals and organizations.

Data leakage detection is crucial in cloud computing environments to prevent such incidents from occurring. The detection of data leakage can be challenging in cloud computing environments due to the distributed nature of cloud computing, where data is stored and processed in various locations. Additionally, the traditional security mechanisms used in non-cloud environments may not be sufficient to protect against data leakage in cloud computing.

In recent years, research has focused on developing new techniques and mechanisms for detecting data leakage in cloud computing environments. These mechanisms utilize various techniques such as static analysis, dynamic analysis, and machine learning algorithms to identify potential data leakage points and detect actual data leakage during runtime.

Static analysis involves analyzing the source code of applications to identify potential data leakage points. Dynamic analysis, on the other hand, involves monitoring the behavior of applications during runtime to detect any actual data leakage. Machine learning algorithms are used to improve the accuracy of data leakage detection by identifying patterns and anomalies in the data that may indicate data leakage.

**2.1 CLOUD COMPUTING**

Cloud computing is the on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user. The term is generally used to describe data centres available to many users over the Internet. A simple definition of cloud computing involves delivering different types of services over the Internet. From software and analytics to secure and safe data storage and networking resources, everything can be delivered via the cloud. You can access it from just about any computer that has internet access.

**2.2TYPES OF CLOUD COMPUTING**

Cloud computing is a broad term which refers to a collection of services that offer businesses a cost-effective solution to increase their IT capacity and functionality. There are three main types of cloud environment, also known as cloud deployment models. The models are.

* **Public cloud**

A public cloud is a type of computing in which a service provider makes resources available to the public via the internet. ... Some public cloud providers offer resources for free, while clients pay for other resources by subscription or a pay-per-usage model. Public clouds are the most common type of cloud computing deployment. Microsoft Azure is an example of a public cloud. In a public cloud, you share the same hardware, storage and network devices with other organisations or cloud “tenants,” and you access services and manage your account using a web browser.

* **Private cloud**

## The private cloud is defined as computing services offered either over the Internet or a private internal network and only to select users instead of the general public. Also called an internal or corporate cloud, private cloud computing gives businesses many of the benefits of a [public cloud](https://azure.microsoft.com/en-in/overview/what-is-a-public-cloud/) - including self-service, scalability and elasticity - with the additional control and customisation available from dedicated resources over a computing infrastructure hosted on-premises. In addition, private clouds deliver a higher level of security and privacy through both company firewalls and internal hosting to ensure operations and sensitive data are not accessible to third-party providers.

## Hybrid cloud

## Hybrid cloud is a solution that combines a private cloud with one or more public cloud services, with proprietary software enabling communication between each distinct service. A hybrid cloud strategy provides businesses with greater flexibility by moving workloads between cloud solutions as needs and costs fluctuate. Hybrid cloud refers to a mixed computing, storage, and services environment made up of on-premises infrastructure, private cloud services, and a public cloud—such as Amazon Web Services (AWS) or Microsoft Azure—with orchestration among the various platforms

## There are three main service models of cloud computing. They are

#### **IaaS (Infrastructure as Service)**

This is the most common service model of cloud computing as it offers the fundamental infrastructure of virtual servers, network, operating systems and data storage drives. It allows for the flexibility, reliability and scalability that many businesses seek with the cloud, and removes the need for hardware in the office. This makes it ideal for small and medium sized organisations looking for a cost-effective IT solution to support business growth. IaaS is a fully outsourced pay-for-use service and is available as a public, private or hybrid infrastructure.

#### **PaaS (Platform-as-a-Service)**

This is where cloud computing providers deploy the infrastructure and software framework, but businesses can develop and run their own applications. Web applications can be created quickly and easily via PaaS, and the service is flexible and robust enough to support them. PaaS solutions are scalable and ideal for business environments where multiple developers are working on a single project. It is also handy for situations where an existing data source (such as CRM tool) needs to be leveraged.

#### **SaaS (Software as a Service)**

This cloud computing solution involves the deployment of software over the internet to various businesses who pay via subscription or a pay-per-use model. It is a valuable tool for CRM and for applications that need a lot of web or mobile access – such as mobile sales management software. SaaS is managed from a central location so businesses don’t have to worry about maintaining it themselves, and is ideal for short-term projects.

**2.3HOW CLOUD STORES THE DATAS?**

The storage outsourcing, the cloud server stores massive data on behalf of its clients (data owners). However, a malicious cloud server can delete some of the client’s data (that are accessed infrequently) to save some space. Secure cloud storage protocols (two-party protocols between the client and the server) provide a mechanism to detect if the server stores the client’s data untampered. Based on the nature of the outsourced data, these protocols are classified as: secure cloud storage protocols for static data (SSCS) and for dynamic data (DSCS) . For static data, the client cannot change her data after the initial outsourcing (e.g., backup/archival data). Dynamic data are more generic in that the client can modify her data as often as needed. In secure cloud storage protocols, the client can audit the outsourced data without accessing the whole data file, and still be able to detect unwanted changes in data done by a malicious server. During an audit, the client sends a random challenge to the server which produces proofs of storage (computed on the stored data) corresponding to that challenge. Secure cloud storage protocols are publicly verifiable if an audit can be performed by any third-party auditor (TPA) using public parameters; or privately verifiable if an auditor needs some secret information of the client. The entities involved in a secure cloud storage protocol and the interaction.

**2.4 NETWORK CODING TECHNIQUES**

In a network coding protocol, each intermediate node (except sender/receiver nodes) on a network path combines incoming packets to output another packet. These protocols enjoy higher throughput, efficiency and scalability than the store-and-forward routing, but they are prone to pollution attacks by malicious intermediate nodes injecting invalid packets. These packets produce more such packets downstream, and the receiver might not finally decode the file sent by the sender node. Secure network coding (SNC) protocols use cryptographic techniques to prevent these attacks: the sender authenticates each packet by attaching a small tag to it. These authentication tags are generated using homomorphic message authentication codes (MACs) or homomorphic signatures. Due to homomorphic property, an intermediate node can combine incoming packets (and their tags) into a packet and its tag in particular, they show that one can exploit some of the algorithms involved in an SNC protocol in order to construct a secure cloud storage protocol for static data. However, their construction does not handle dynamic data — that makes it insufficient in many applications where a client needs to update (insert, delete or modify) the remote data efficiently. Further investigations are needed towards an efficient DSCS construction using a secure network coding (SNC) protocol.

Network coding techniques have been used to construct distributed storage systems where the client’s data are disseminated across multiple servers. However, they primarily aim to reduce the repair bandwidth when some of the servers fail. On the other hand, we explore whether we can exploit the algorithms involved in an SNC protocol to construct an efficient and secure cloud storage protocol for dynamic data (for a single storage server). Although dynamic data are generic in the sense that they support arbitrary update (insertion, deletion and modification) operations, append-only data (where new data corresponding to a data file are inserted only at the end of the file) find numerous applications as well. These applications primarily maintain archival as well as current data by appending the current data to the existing datasets. Examples of append-only data include data obtained from CCTV cameras, ledgers containing monetary transactions, medical history of patients, data stored at append-only databases, and so on. Append-only data are also useful for maintaining other log structures (e.g., certificates are stored using append-only log structures in certificate transparency schemes). In many of such applications, the data owner requires a cloud server to store the bulk data in an untampered and retrievable fashion with append being the only permissible update. Although secure cloud storage schemes for generic dynamic data also work for append-only data, a more efficient solution (specific to append-only data files) would be helpful in this scenario.

**2.5 LITERATURE SURVEY**

**1.Author:** B. Sengupta and S. Ruj, 2016

**Title:** “Publicly verifiable secure cloud storage for dynamic data using secure network coding,

**Description:** Cloud service providers offer storage outsourcing facility to their clients. In a secure cloud storage (SCS) protocol, the integrity of the client's data is maintained. In this work, we construct a publicly verifiable secure cloud storage protocol based on a secure network coding (SNC) protocol where the client can update the outsourced data as needed. To the best of our knowledge, our scheme is the first SNC-based SCS protocol for dynamic data that is secure in the standard model and provides privacy-preserving audits in a publicly verifiable setting. Furthermore, we discuss, in details, about the (im)possibility of providing a general construction of an efficient SCS protocol for dynamic data (DSCS protocol) from an arbitrary SNC protocol. In addition, we modify an existing DSCS scheme (DPDP I) in order to support privacy-preserving audits. We also compare our DSCS protocol with other SCS schemes (including the modified DPDP I scheme). Finally, we figure out some limitations of an SCS scheme constructed using an SNC protocol.

**2.Author:** S. Agrawal and D. Boneh.  2009

**Title:** MAC-based integrity for network coding. In Applied Cryptography and Network Security

**Description:** Network coding has been shown to improve the capacity and robustness in networks. However, since intermediate nodes modify packets en-route, integrity of data cannot be checked using traditional MACs and checksums. In addition, network coded systems are vulnerable to pollution attacks where a single malicious node can flood the network with bad packets and prevent the receiver from decoding the packets correctly. Signature schemes have been proposed to thwart such attacks, but they tend to be too slow for online per-packet integrity.

Here we propose *a*homomorphic MAC which allows checking the integrity of network coded data. Our homomorphic MAC is designed as a drop-in replacement for traditional MACs (such as HMAC) in systems using network coding.

**3.Author:** Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li,2011

**Title:** “Enabling public auditability and data dynamics for storage security in cloud computing,

**Description:** Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. It moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. This unique paradigm brings about many new security challenges, which have not been well understood. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of the client through the auditing of whether his data stored in the cloud is indeed intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamics via the most general forms of data operation, such as block modification, insertion and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or backup data only. While prior works on ensuring remote data integrity often lacks the support of either public auditability or dynamic data operations, this paper achieves both. We first identify the difficulties and potential security problems of direct extensions with fully dynamic data updates from prior works and then show how to construct an elegant verification scheme for the seamless integration of these two salient features in our protocol design. In particular, to achieve efficient data dynamics, we improve the existing proof of storage models by manipulating the classic Merkle Hash Tree construction for block tag authentication. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis show that the proposed schemes are highly efficient and provably secure.

**4.Author:** G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik,2009

**Title:** Scalable and efficient provable data possession

**Description:** Storage outsourcing is a rising trend which prompts a number of interesting security issues, many of which have been extensively investigated in the past. However, Provable Data Possession (PDP) is a topic that has only recently appeared in the research literature. The main issue is how to frequently, efficiently and securely verify that a storage server is faithfully storing its client's (potentially very large) outsourced data. The storage server is assumed to be untrusted in terms of both security and reliability. (In other words, it might maliciously or accidentally erase hosted data; it might also relegate it to slow or off-line storage.) The problem is exacerbated by the client being a small computing device with limited resources. Prior work has addressed this problem using either public key cryptography or requiring the client to outsource its data in encrypted form.

In this paper, we construct a highly efficient and provably secure PDP technique based entirely on symmetric key cryptography, while not requiring any bulk encryption. Also, in contrast with its predecessors, our PDP technique allows outsourcing of dynamic data, i.e, it efficiently supports operations, such as block modification, deletion and append.

**5.Author:** K. Omote and T. T. Phuong, 2015

**Title:** “DD-POR: Dynamic operations and direct repair in network coding-based proof of retrievability,”

**Description**: POR (Proof of Retrievability) is a protocol by which clients can distribute their data to cloud servers and can check if the data stored in the servers is available and intact. Based on the POR, the network coding is applied to improve network throughput. Although many network coding-based PORs have been proposed, most of them have not achieved the following practical features: direct repair and dynamic operations. In this paper, we propose the DD-POR (Dynamic operations and Direct repair in network coding-based POR) to address these shortcomings. When a server is corrupted, the DD-POR can support the direct repair in which the data stored in the corrupted server can be repaired using the data provided directly from the healthy servers. The client is thus free from the burden of data repair. Furthermore, the DD-POR allows the client to efficiently perform dynamic operations, i.e., modification, insertion and deletion.