PROVABLE DATA POSSESSION (PDP) AND PROOFS OF RETRIEVABILITY (POR) OF CURRENT BIG USER DATA

Cryptographic schemes in cloud storage

By

Ieuan Walker

A thesis presented for the degree of Master of Philosophy
Cardiff Metropolitan University

Director of studies: Dr. Chaminda Hewage
Supervisors: Dr. Ambikesh Jayal
Mr. Allan Hagan

Cardiff Metropolitan University | Prifysgol Metropolitan Caerdydd

Acknowledgement

Thank you to my supervisor, Dr. Chaminda Hewage, for providing valuable guidance and feedback throughout this project.

I would also like to thank Dr. Ambikesh Jayal for contacting me about this amazing opportunity and helping me along the way.

And finally, I would like to thank everyone at KESS II and Mr. Allan Hagan from Ultranyx, without their support this opportunity would not have been possible.

Contents

Acronyms

Acronym	Definition
AES	Advance encryption Standard
AWS	Amazon Web Services
BLS	Boneh–Lynn–Shacham
CISO	Chief Information Security Officer
CSA	Cloud Security Alliance
DES	Data Encryption Standard
DPDP	Dynamic provable data possession
DPDP	Dynamic Provable Data Possession
ENISA	European Network and Information Security Agency
EU	European Union
HP	Hewlett Packard
IAAS	Infrastructure as a Service
ISSA	Information Systems Security Association
IT	Information Technology
KESS	Knowledge Economy Skills Scholarships
MAC	Message authentication code
NIST	National Institute of standard and technology
PAAS	Platform as a Service
PDP	Provable data possession
POR	Proofs of Retrievability
RSA	Rivest–Shamir–Adleman
SAAS	Software as a Service
SSL	Secure Sockets Layer
STAR	Security, Trust & Assurance Registry
TLS	Transport Layer Security
US	United States
VPN	Virtual private networks

Abstract

Background: A growing trend over the last few years is storage outsourcing. Where the concept of third-party data warehousing has become more popular. This trend prompts serval interesting security issues. One of the biggest issues with third-party data providers is accountability. This thesis, reviews two schemas/ algorithms that allows users to check the integrity and availability of their outsourced data on untrusted data stores. The reviewed schemas are **P**rovable **D**ata **P**ossession (PDP) and **P**roofs **o**f **R**etrievability (POR). Both are cryptographic protocols designed to give clients the assurance needed that their data is secure on the untrusted file storage. Furthermore, a conceptual framework is proposed to mitigate the weaknesses of the current storage solutions.

Results: PDP and POR schemas do have different responsibilities. PDP main aim is to ensure that the client file is intact and has not been tampered, whereas POR main aim is to guarantee that the client can retrieve the file even with small file corruption. Both have real-world usage and is critical for today's data-centric world.

Conclusions: The differences between PDP and POR schemas are becoming less with each new iteration. This thesis demonstrates the difference and weaknesses in both approaches and provides a conceptual model as a building block to further the research in this area.

Chapter 1: Introduction

A growing trend over the last few years is storage outsourcing, where the concept of third-party data warehousing has become more popular. What is data outsourcing? It essentially means that the data owner moves its data to a third-party storage provider, in exchange for a fee. There are a lot of appealing features to outsourcing your data such as reduced cost, more space, less complexity in operations, reliability, and support (Carroll, 2019). Many organisations are migrating their data to these third-party data providers, for the many benefits that they provide.

The data outsourcing trend prompts several interesting security issues. One of the biggest issues with third-party data providers is accountability. If the client rarely accesses the data stored on the untrusted provider, how can the client be assured that their data is still being stored and have not been tampered with? For example, if the storage provider experiences a hardware failure and causes some clients to lose their data, there could be a situation where the storage provider would not notify the clients as there is a chance that the data will never be accessed again, meaning the client will never find out. However, Dodis (2009) stated: "a malicious storage provider might even choose to delete rarely accessed files to save money".

Therefore, it is important for the clients to checks their data regularly, to make sure that their data has not been tampered with or deleted. But in the majority of cases, there could be a huge amount of data/files, that it will be infeasible for the client to download the entire file to validate it, with constraints such as bandwidth and time, especially if the files need frequent validation.

Research into cloud storage providers first started with data authentication and integrity, meaning, how to efficiently and securely return the complete and correct response to a client's query – *'authenticity and non-repudiation of the answer to a database query'* (Devanbu, et al., 2002). Then focus shifted to, how to query encrypted data efficiently. In a study titled 'Secure Conjunctive Keyword Search over Encrypted Data' they researched when a *"user stores encrypted documents (e.g. e-mails) on an untrusted server. In order to retrieve documents satisfying a certain search criterion, the user gives the server a capability that allows the server to identify exactly those documents"* (Golle, et al., 2004).

This thesis will be reviewing two schemas/ algorithms that allows users to check the integrity and availability of their outsourced data on untrusted data stores. The schemes that will be reviewed are **P**rovable **D**ata **P**ossession (PDP) and **P**roofs **o**f **R**etrievability (POR); both are cryptographic protocols designed to give clients the assurance needed that their data is secure on the untrusted file storage.

The main issue that these schemes attempt to solve is how to frequently, efficiently and securely verify that the client data is intact and able to retrieve the file if needed even if the file is corrupt, they can do this verification without needed to check the entire file.

This thesis is separated into four stages. The first being, discussing cloud service providers in general, then discussing Provable Data Possession (PDP), Proofs of Retrievability (POR), create and analyse a conceptrion model based on the previous finding, and end with a conclusion summarising the finding.

This research aims to identify the current algorithms in use in the cloud storage space. The two algorithms this thesis will critically analyse are **P**rovable **D**ata **P**ossession (PDP) and **P**roofs **o**f **R**etrievability (POR). The reasons these algorithms were chosen are because they are closely related to each other and are critical in today's data-centric world. Once the above two algorthrims have been analised I will then look at the next steps for these algorthms and create a conceptual model that could help as a basis for the next version of these algorithms.

This research will consist of 3 objectives:

1. Conduct a thorough Literature Review on Provable Data Possession (PDP) and Proofs Of Retrievability (POR).
2. Critically analyse the literature around provable data possession (PDP) and proofs of retrievability (POR); this objective aims to identify the strengths and weaknesses of both algorithms and gain a better understanding of both approaches.
3. Using the research and knowledge gained from the above two objectives, this thesis will then proposes a conceptual model based on the strength and weaknesses identified of modern cloud storages.

The research consists of analysing different approaches reported in the recent literature and determining the differences between the approaches. This is achieved by comparing and contrasting different Key Performance Indicators (KPIs) generated by each approach, and determining which approach is better under different circumstances. Furthermore, a thorough critical review of literature is carried out. This includes both academic publications, patents, white papers and relevant case studies. Each source of the literature will be discussed in chronological order, to understand the progression of the approaches over time. By doing this will be answering my first and second which is to '*identify the strengths and weaknesses of both algorithms'* and *'gain a better understanding of both approaches*'.

The research will be basing conclusions on the facts and figures collected. The type of data that will be collected can be based on server performance, latency or size of the data payloads. These factors will be used to propose a conceptual model for provable and retrievable cloud data (Research objective 3). This model can then be used as a basis of further research in this area.

Chapter 2: Literature review

There are a number of different ways to explain cloud computing but the simplest and most comprehensive definition can be found in the Control Engineering journal by Brandl Dennis (2010), he describes cloud computing as a *"collections of IT resources (servers, databases, and applications) which are available on an on-demand basis, provided by a service company, available through the internet, and provide resource pooling among multiple users."*

2.1. Cloud Architecture

Before researching into a small specific area of cloud storage it is important to understand the architecture patterns currently in use and previous methods that have evolved over the years.

The concept of cloud computing was first conceived in 1961 by a man called McCarthy and explored further by Licklider in 1963. The implementation of cloud computing started to build up in the 1990s with the introduction of the internet (Alali & Yeh, 2012), this is when it was possible to remotely connect to a computer and exchange information as well as using remote applications; however, it was not until the 2000s when Web 2.0 was released that cloud computing could share information globally.

Common Traits

For organizations to use new technologies effectively, such as the cloud they must understand exactly what it consists of. The National Institute of Standards and Technology has issued a publication called 'The NIST Definition of Cloud Computing'.

The report outlines 5 essential characteristics, these are -

On-demand Self-service:

Clients can add and remove computing capabilities (server time/ network storage) automatically without the need to contact the service provider (Mell & Grance, 2011).

Broad network access:

"Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations)" (Mell & Grance, 2011).

Resource pooling:

Mell & Grance (2011) describes resource polling as the service providers resources are shared among its multiple consumers and can be reassigned based on demand, allowing consumers to scale as needed and only using the resource that is needed at that current time. All this is done without any consumers seeing a difference in its performance.

Rapid Elasticity:

"*Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time"* (Mell & Grance, 2011).

Measured Service:

"Cloud systems automatically control and optimize resource use by leveraging a metering capability1 at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service." (Mell & Grance, 2011)

"When agencies or companies use this definition, they have a tool to determine the extent to which the information technology implementations they are considering meet the cloud characteristics and models" (National Institute of Standards and Technology, 2011) says NIST computer scientist Peter Mell who co-authored the report.

The purpose of this publication is to help system planners, program managers, technologists, and others (Mell & Grance, 2011) to understand the different aspects of cloud providers and to serve as a means for broad comparisons of cloud services.

Actors involved

NIST has defined five roles in the NIST's Cloud Computing Reference Architecture (Lui, et al., 2011), these are –

Cloud Consumer

A person or organization that maintains a business relationship with, and uses service from, Cloud Providers.

Cloud Provider

A person, organization, or entity is responsible for making a service available to interested parties; this is the entity responsible for creating and developing cloud services and making them available to interested parties. Examples of cloud providers are Microsoft with 'Azure', Amazon with 'Amazon Web Service' (AWS) and Google with 'Google Cloud Platform'.

Cloud Auditor

A party that can conduct an independent assessment of cloud services, information system operations, performance and security of the cloud implementation.

Cloud Broker

An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.

Cloud Carrier

An intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.

Here is a diagram from the publication which demonstrates the interactions between each of the actors.

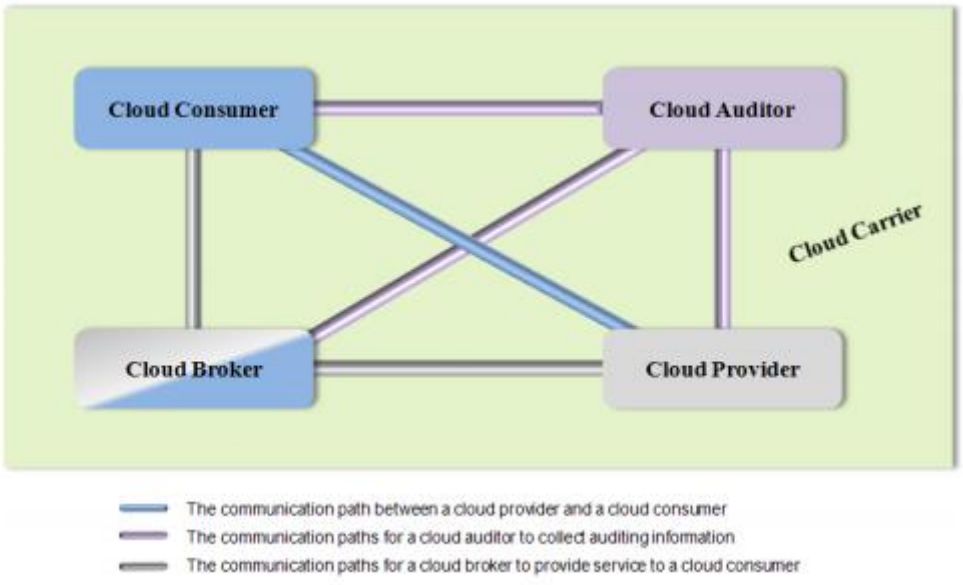


Figure 1 - Interactions between the Actors in Cloud Computing (Mell & Grance, 2011)

The publication continues by running usage scenarios of the above diagram, explain why each actor would interact with another. NIST continues by discussing each actor in detail explaining their role, responsibilities and what each actor need to complete their role.

Deployment models

NIST identified four different deployment models of the cloud, "*Cloud infrastructure may be operated in one of the following deployment models: public cloud, private cloud, community cloud, or hybrid cloud. The differences are based on how exclusive the computing resources are made to a Cloud Consumer*" (Lui, et al., 2011). But after this article was published a new deployment model has emerged called multi-cloud.

Public Cloud

Public cloud is owned by an organisation selling cloud services to the public or a large industry group (Alali & Yeh, 2012). In a public cloud infrastructure, services, storage, applications and services are pooled together and shared to multiple users (Joshi, 2012)

Private Cloud

This is a deployment model that involves a distinct and secure cloud environment where only a specified client can operate (Interoute, 2019). Private clouds allow the organisation to provision equipment and resources to be used exclusively by them (Mell & Grance, 2011) and are usually managed via internal resources.

Community Cloud

A community cloud is far less common than the other types. The definition of the community cloud deployment model from Peter Mell and Timothy Grance who are researchers at NIST. "*The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises."* (Mell & Grance, 2011)

Hybrid Cloud

A Hybrid Cloud is the mixture of two or more distinct (private or public) cloud infrastructures (Mell & Grance, 2011), allowing data and applications to be shared between them (Microsoft, 2019). This means that all the environments are "*bound together by standardized or proprietary technology that enables data and application portability*". This approach is commonly used by large companies to gradually move their infrastructure to the cloud without a rush or a large capital investment.

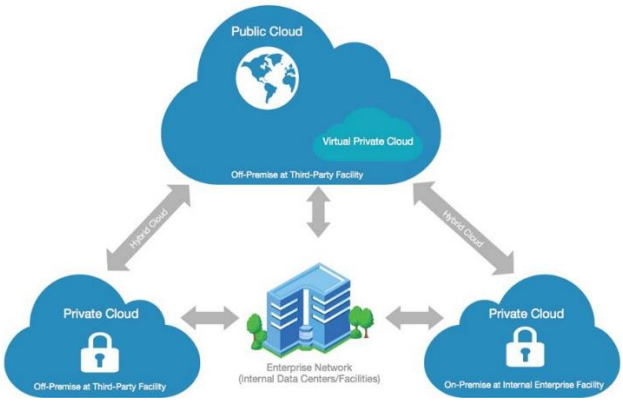


Figure 2 - Hybrid cloud diagram

Multi-cloud

HP describes multi-cloud as *"the use of multiple cloud computing services in a single heterogeneous architecture to reduce reliance on single vendors, increase flexibility through choice, mitigate against disasters, and so on."* (Ko & Boutelle, 2018). NIST has described four different cloud deployment models, which are private cloud, community cloud, public cloud and hybrid cloud (Mell & Grance, 2011). So, what is multi-cloud? Well, people have been mistakenly been using multi-cloud and hybrid cloud interchangeably. Ko and Boutella (2018) continue by explaining *"Multicloud differs from hybrid cloud in that it refers to multiple cloud services rather than multiple deployment modes (public, private, and legacy). Multicloud uses multiple cloud providers (Amazon Web Services, Azure, internal IT, and so on) for multiple workloads".*

Table 1 - Key advantages and characteristics of each deployment model demonstrates the key advantages and characteristics of each deployment model.

Private	Public	Hybrid	Multi
Consists of computing resources used exclusively by one business or organisation	Resources (like servers and storage) are owned and operated by a third-party cloud service provider and delivered over the Internet	Combine on-premises infrastructure, or private clouds, with public clouds so that organisations can reap the advantages of both	The use of two or more cloud computing services
Can be physically located at your organisation's on-site data centre, or it can be hosted by a third-party service provider	All hardware, software and other supporting infrastructure are owned and managed by the cloud provider	Data and applications can move between private and public clouds for greater flexibility and more deployment options	Avoid vendor lock-in
Services and infrastructure are always maintained on a private network and the hardware and software are dedicated solely to your organisation	You share the same hardware, storage and network devices with other organisations or cloud "tenants"	In a hybrid cloud, "cloud bursting" is also an option. This is when an application or resource runs in the private cloud until there is a spike in demand (such as a seasonal event like online shopping or tax filing), at which point the organisation can "burst through" to the public cloud to tap into additional computing resources.	Can refer to any implementation of multiple software as a service (SaaS) or platform as a service (PaaS) cloud offerings, today, it generally refers to a mix of public infrastructure as a service (IaaS) environments, such as Amazon Web Services and Microsoft Azure
Can make it easier for an organisation to customise its resources to meet specific IT requirements	Lower costs – no need to purchase hardware or software, and you only pay for the service you use.	Control – your organisation can maintain a private infrastructure for sensitive assets	Find the optimal cloud service for a business or technical need
Are often used by government agencies, financial institutions and any	No maintenance – your service provider provides the maintenance.	Flexibility – you can take advantage of additional resources in the public cloud	Increased redundancy

other medium to large-sized organisations with business-critical operations seeking enhanced control over their environment		when you need them	
More flexibility – your organisation can customise its cloud environment to meet specific business needs	Near-unlimited scalability – on-demand resources are available to meet your business needs.	Cost-effectiveness – with the ability to scale to the public cloud, you pay for extra computing power only when needed	Prevent data loss or downtime due to a localized component failure in the cloud
Improved security – resources are not shared with others, so higher levels of control and security are possible	High reliability – a vast network of servers ensures against failure.	Ease – transitioning to the cloud does not have to be overwhelming because you can migrate gradually – phasing in workloads over time	More price-competitive cloud services or taking advantage of the speed, capacity or features offered by a cloud provider in a geography
High scalability – private clouds still afford the scalability and efficiency of a public cloud			Ability to select different cloud services or features from different providers. This is helpful since some cloud environments are better suited than others for a task

Table 1 - Key advantages and characteristics of each deployment model

Cloud computing Structures

Cloud computing has three main structures, these are –

Infrastructure as a Service (IaaS) "*is the delivery of hardware (server, storage and network), and associated software (operating systems virtualization technology, file system), as a service."* (Bhardwaj, et al., 2010)

Platform as a Service (PaaS) *"offers a high-level integrated environment to build, test and deploy custom applications."* (Albor, et al., 2015). *"Platform as a service (PaaS) is a complete development and deployment environment in the cloud, with resources*

that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications. You purchase the resources you need from a cloud service provider on a pay-as-you-go basis and access them over a secure Internet connection." (Microsoft, 2018)

Software as a Service (SaaS) "is a licensing and delivery model whereby software is centrally managed and hosted by a provider and available to customers on a subscription basis" (Amazon, 2018)

Each of these structures is built on top of each other with IaaS giving organisations almost complete control over the hardware and SaaS limiting the control the user has.

Risks

There are many risks associated with migrating infrastructure and data to a third-party cloud provider, such as lock-in, legislation and jurisdiction, and data protection.

2.2. Cloud Security Organization and Agencies

Since the popularity of cloud service providers has grown, several private and government-led agencies have emerged to guide cloud service providers and users into best practices around cloud security. The main goal of these cloud security agencies is to promote cloud security research and advance security within the cloud.

Cloud Security Alliance (CSA)

Cloud Security Alliance also known as CSA is a non-profit organization (Techopedia, 2018). The idea of the CSA originated in November 2008 from Jim Reavis during his presentation at the ISSA CISO Forum in Las Vegas. Where a series of meetings continued with industry leaders and in December 2008 CSA was formalised and founded (Cloud Security Alliance, 2018a). The CSA is supported by several large IT companies such as Microsoft, Amazon, Oracle and many more (Cloud Security Alliance, 2018b).

The main goal of the CSA is to provide security assurance and education within the field of cloud security. They achieve this by operating a popular cloud security provider certification program, called the 'CSA Security, Trust & Assurance Registry (STAR)', STAR consist of 3 different tiers, Self-assessment/ 3rd party certification and continuous auditing (Cloud Security Alliance, 2018c). More on this program can be found on the CSA official website.

National Institute of Standards and Technology (NIST)

National Institute of Standards and Technology, also known as NIST, is a government-funded organisation in the US.

National Institute of Standards and Technology is a government-funded organization in the US, continuously assisting cloud computing platform users by identifying security-related vulnerabilities in the platform. Security issues discussed by NIST are specifically focused on public cloud vendors, as it states that organizations have more control of each layer of security when a private cloud deployment model is used.

European Network and Information Security Agency (ENISA)

The European Network and Information Security Agency is another government-funded organization aiming to provide better security functionality in the cloud computing platform.

ENISA published its first document "Cloud Computing Benefit, Risk and Recommendation for Information Security" in November 2009. The document began by highlighting the key benefits of security for cloud computing platforms. *"ENISA is actively contributing to European cybersecurity policy, supporting Member States and European Union stakeholders to support a response to large-scale cyber incidents that take place across borders in cases where two or more EU Member States have been affected. This work also contributes to the proper functioning of the Digital Single Market."* (European Union Agency for Cybersecurity (ENISA), 2019)

2.3 Data Encryption/ integrity

Data encryption is very important. The purpose of data encryption is

Two types of encryption, Symmetric and Asymmetric

There are two types of encryption, these are symmetric and asymmetric encryption. *"A symmetric key, or secret key, uses one key to both encode and decode the information. This is best used for one to one sharing and smaller data sets. Asymmetric, or public key cryptography, uses two linked keys – one private and one public. The encryption key is public and can be used by anyone to encrypt. The opposite key is kept private and used to decrypt."* (Forcepoint, n.d.)

Advance encryption Standard (AES)

AES is one of the most popular and widely adopted symmetric encryption algorethims currently in use.

Before AES, Data Encryption Standard (DES) was used. It was replaced by AES as the key size for DES was too small and as computers got more powerful, it was considered vulnerable against exhaustive key search attacks (brute force attacks) (Quisquater & Fran¸cois-Xavier, 2005).

AES has three block ciphers to choose from, AES-128, AES-192 and AES-256. The only difference between each is a larger key size is used to encrypt. AES-128 is sufficient for most cases as currently it still protects against exhaustive key search attacks. AES-256 is generally used by the milliteray and government to protect agains sensitive data. The larger the key size used the *"more processing power and can take longer to execute. When power is an issue -- particularly on small devices -- or where latency is likely to be a concern, 128-bit keys are likely to be a better option"* (Cobb, n.d.)

Rivest–Shamir–Adleman (RSA)

RSA was first described in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman of the Massachusetts Institute of Technology (Cobb, n.d.). RSA is a asymmetric cryptography scheme.

What makes RSA different from other asymmetric encryption is that both the public key and the private key can be used to encrypt the message, the oppiste key is then used to decrypt it. *"This attribute is one reason why RSA has become the most widely used asymmetric algorithm: It provides a method to assure the confidentiality, integrity, authenticity, and non-repudiation of electronic communications and data storage."* (Cobb, n.d.)

There are many variants of RSA, such as, Shared RSA, Multiprime RSA, Common Prime RSA, etc. The reason for all these variants is because there are a number of weakness in the original design, for example (Balasubramanian, 2014) –

- High computational cost
- Slow
- If the private key is lost then all received message cannot be decrypted

AES vs RSA

AES is prodominetly used to protect data at rest, for example, encryptin databases, file systems etc. Whereas RSA is often used across the interent securing connections, i.e. to virtual private networks (VPN), websites via Secure Sockets Layer (SSL)/ transport Layer Security (TLS).

As you can see both cryptoghray methods has it's uses, and more often than not are used together to get better performance and security (Townsend, 2019)

Chapter 3: Methodology

This chapter looks at various research methodologies and research methods that are commonly used in the field of information systems. Several approaches are implemented in this project to complete the aims and objectives of the research and will be discussed and justified in the following chapter.

Research philosophy interprets the source, nature and development of knowledge which defines the ways in which data about the research problem is collected, analyzed and utilized. In order to answer the research questions in this thesis, secondary data was collected from the academic sources and critically analyzed to answer the research question to generate new knowledge. The interpretivism research philosophy is followed throughout this research to investigate the research question in depth. Since this research aims to find a solution for specific problem(s), action research is followed to find valuable outcomes for practical problems. Furthermore, inductive research methodology has been followed since the specific research question(s) were formulated in the beginning of the research process (see Chapter 1).

3.1 Data collection

The data collection methodology followed in this thesis is elaborated below. The review is carried out using the publicly available, secondary data sources which discuss different aspects of validating and recovering data from an untrusted provider. The main data sources used in this review are SCOPUS library, Web of Science (WoS) citation database, ACM library, IEEE Xplorer, Google Scholar, Researchgate, etc. A number of keyword searches were used to find relevant studies and reviews necessary to answer the research questions of this thesis. The main keywords combinations included “Data integratiy”, “Cloud storage”, “data retrievability”, “validating data”, and other relevant key words. An exclusion criterion was not used.

3.2 Data analysis

As for the schemas, they were selected for an analytical review based on the number of references found based on all the keyword combinations. In addition to the above keyword

search by the authors, recommendations by previously published research, tutorials, surveys and reviews were used to select the schemas to focus on this review. The schemas have been analyzed, discussed and summarized. The academic papers from the literature for each schema were ordered chronologically, and where chosen due to the contribution they made to the overall schema, and where found due to the amount of journal papers that referenced them.

3.3 Data access

The research outcomes and results of this thesis will be available for academic audience through the Dspace research sharing platform of Cardiff Metropolitan University. The original work of this thesis should be duly cited in future research by the respective academics and researchers.

Chapter 4: Provable data possession

Provable data possession is a way to give the tenants a means to verify that their data, stored at untrusted storage is intact and has not been tampered with, without requiring the tenant to download the actual data.

A brief general overview of the PDP model is that the client pre-processes the data and sends it to and untrusted data store, while only keeping a small amount of metadata to use later. The client can then later ask the storage provider to prove that the data they sent has not been tampered with or deleted. All without requiring the file to be downloaded. (Erway, et al., 2009)

4.1. Provable Data Possession at Untrusted Stores (Guiseppe, et al., 2007)

In 2007 (Guiseppe, et al., 2007) presented a new type of scheme that they called '**P**rovable **D**ata **P**ossession' or PDP for short. There proposed scheme allows tenants who have stored files on an untrusted storage a means of verifying that their data is intact and has not been tampered with without requiring to download the actual file.

The main goal of this scheme is to be able to check the integrity of files as quickly as possible. It does this by using a minimal amount of metadata (*which the tenants stores*).

"The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely distributed storage systems" (Guiseppe, et al., 2007)

Figure 3 - PDP pre-process and store diagramand Figure 4 - PDP verify server possession diagramdemonstrate how this schema works. Figure 3 - PDP pre-process and store diagramshows the process of sending a file to the untrusted source, and Figure 4 - PDP verify server possession diagram shows how to verify the file on the server.

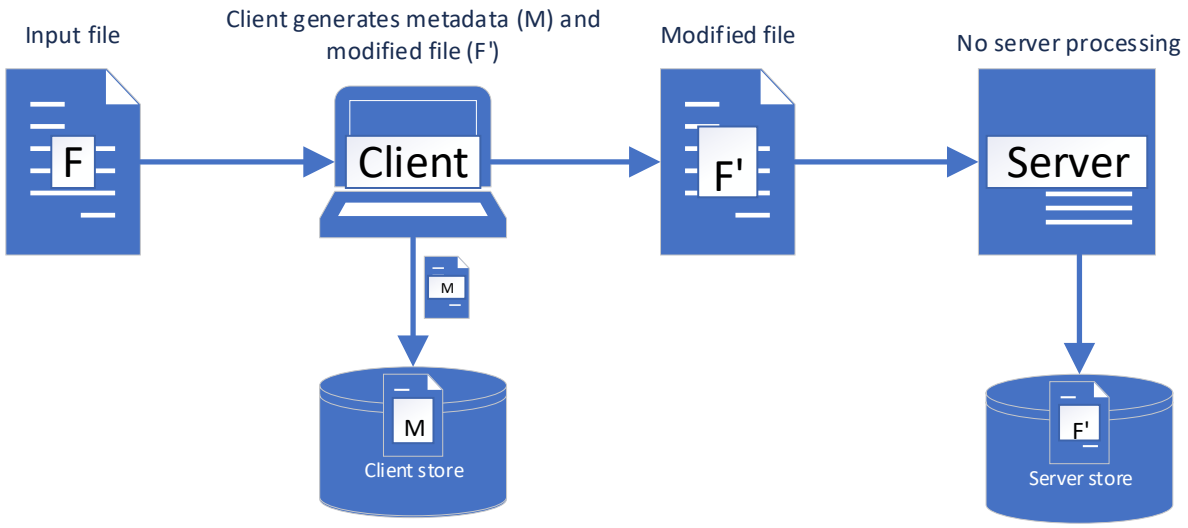


Figure 3 - PDP pre-process and store diagram

The first part of this schema is sending the file to the server. Before uploading the file 'F' to the 'server store' the client calculates the metadata called *'homomorphic verifiable tags'*. These tags are "*computed for multiple file blocks can be combined into a single value*" (Guiseppe, et al., 2007). The client creates *'homomorphic verifiable tags'* for each file block using its private and public keys. The client then stores these tags to verify the file later 'M'. The file, metadata, tags, and the public key are then sent to the server store 'F[1]'. These are all used later when the client challenges the server later. The added tags are considerably smaller than the file itself leaving the overhead on the storage server to a minimum.

Once the file has been sent to the server-client then discard the file and the tags, and only keeps the metadata 'M' and the public/ private keys.

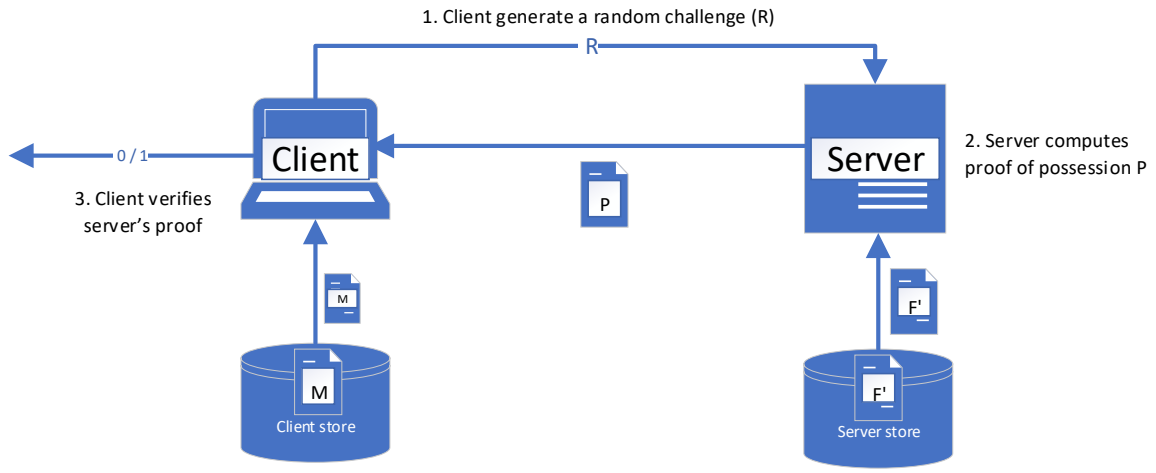


Figure 4 - PDP verify server possession diagram

The second part of the scheme is the process of verifying the server has possession of the file. "*The client can verify that the server possesses the file by generating a random challenge against a randomly selected set of file blocks. Using the queried blocks and their corresponding tags, the server generates a proof of possession."* (Guiseppe, et al., 2007) The server then returns the generated proof 'P' to the client. The client then verifies the response and *"The client is thus convinced of data possession, without actually having to retrieve file blocks"* (Guiseppe, et al., 2007).

In their paper, they reviewed previous work on similar protocols but found that they had several drawbacks such as –

- Require expensive service computation or communication over the entire file (Filho & Barreto, 2006) and (Deswarte, et al., 2004).
- Linear storage for the client.
- Do not provide security guarantees for data possession.

The main drawback to the PDP scheme proposed is that it only applies to static data. Which means if the client wishes to modify the data, they will have to run through the PDP scheme again from the start.

4.2. Scalable and Efficient Provable Data Possession (Ateniese, et al., 2008)

The PDP scheme mentioned above (Guiseppe, et al., 2007) is perfect for static data that achieves O(1) for verification. Which means the size of the file does not affect the time to verify the file. But while static models are good for some use case such as libraries and scientific datasets, it is important to consider dynamic cases which would allow inserting, modifying or deleting files. This is essential for –

- Practical cloud file storage. A paper titled 'Scalable secure file sharing on untrusted storage' uses the PDP schema to create a *"cryptographic storage system that enables secure file sharing without placing much trust on the file servers. In particular, it makes novel use of cryptographic primitives to protect and share files. Plutus features highly scalable key management while allowing individual users to retain direct control over who gets access to their files"* (Kallahalla, et al., 2003)

- Databases. *"Some emerging applications require programs to maintain sensitive state on untrusted hosts"* (Maheshwari, et al., 2000). In the paper they described the PDP process and how they created an encrypted database using it – *"The database is encrypted and validated against a collision-resistant hash kept in trusted storage"* (Maheshwari, et al., 2000).
- Peer-to-peer storage. *"OceanStore is a utility infrastructure designed to span the globe and provide continuous access to persistent information. Since this infrastructure is comprised of untrusted servers, data is protected through redundancy and cryptographic technique"* (Kubiatowicz, et al., 2000)

Ateniese, et al., (2008) has developed a dynamic PDP schema called Scalable PDP. That allows somewhat limited dynamic data, meaning it enables appending, modifying and deleting blocks but does not allow inserting blocks.

Their scheme *"consists of two phases: setup and verification (also called challenge in the literature)"* (Ateniese, et al., 2008), much like Guiseppe, et al., (2007) schema. But the new twist that the authors added to the PDP area is the idea of creating all future challenges during the setup phase and then store the pre-computed answers as metadata on the client (Erway, et al., 2009)**.** *"the owner OWN generates in advance t possible random challenges and the corresponding answers"* (Ateniese, et al., 2008)**.**

Due to this approach, it limits the number of updates and challenges the client can perform. It also has a side effect of preventing the possibility of block insertions anywhere, and only allows the clients to append its blocks.

The authors recognise this limitation by stating *"one potentially glaring drawback of our scheme is the prefixed (at setup time) number of verifications t"* (Ateniese, et al., 2008)**.** They go on to say that the only way to increase the number of challenges and update would be by running through the setup phase again, requiring the client (OWN) to retrieve the entire file (D) from the server (SVR). But this approach would be problematic and impractical for large files.

But they go on to justify this decision by assuming *"that OWN wants to periodically (every M minutes) obtain a proof of possession and wants the tokens to last Y years. The number of verification tokens required by our scheme is thus: (Y × 365× 24× 60)/M. The graph in Figure*

1 shows the storage requirements (for verification tokens) for a range of Y and M values. The graph clearly illustrates that this overhead is quite small" (Erway, et al., 2009).

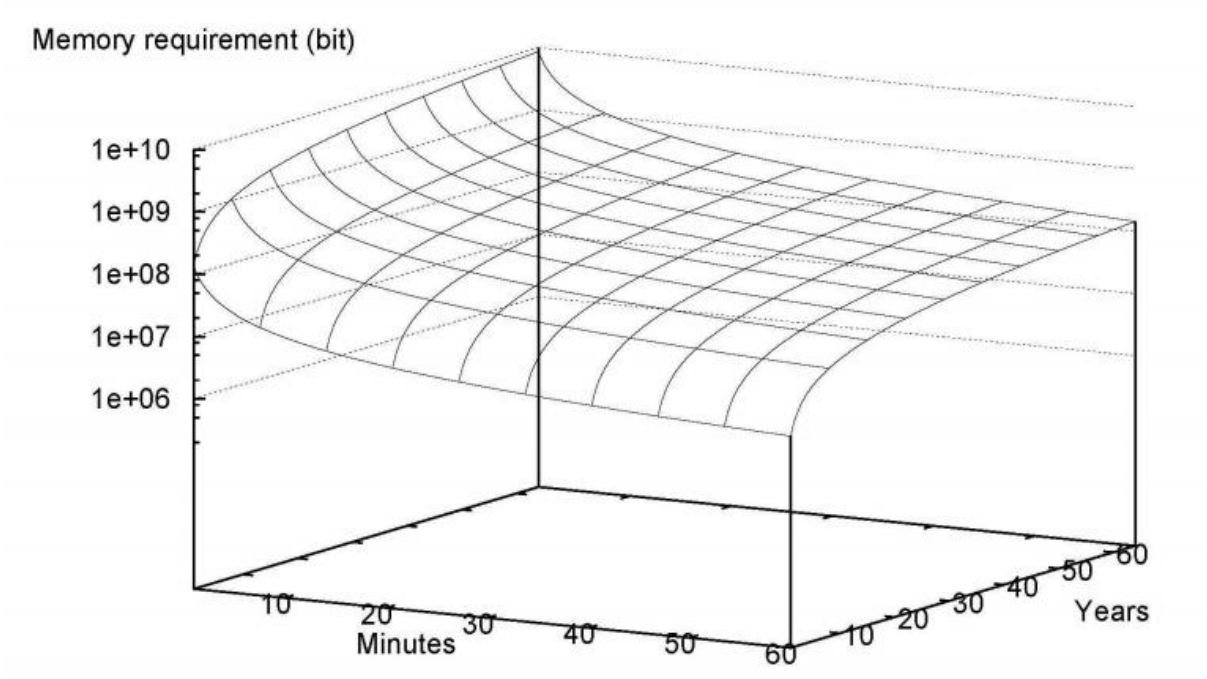


Figure 5 - Token Storage Overhead (in bits): X-axis shows verification frequency (in minutes) and Y-axis

4.3. Dynamic Provable Data Possession (Erway, et al., 2009)

Erway, et al., (2009) released a paper in 2009 titled 'Dynamic provable data possession' (DPDP) which was built upon the provable data possession model (PDP), which extends the functionality of it to support deletion, modification and insertion of data.

Their work was released shortly after Ateniese, et al., (2008). Both papers built upon the earlier work of Guiseppe, et al., (2007). But the main difference between the scalable PDP (Ateniese, et al., 2008) and the DPDP (Erway, et al., 2009) is that Ateniese, et al., (2008) uses a random oracle model whereas DPDP scheme is "*provably secure in the standard model"* (Erway, et al., 2009). They go on to demonstrate the differences by creating a table comparing PDP (Guiseppe, et al., 2007), scalable PDP (Ateniese, et al., 2008), DPDP I, DPDP II (Erway, et al., 2009).

Scheme	Server comp	Client comp	Comm	Model	Block operations				Probability of detection
					append	modify	insert	delete	
PDP	O(1)	O(1)	O(1)	RO	✓				1 – (1 – f) C

Scalable PDP	O(1)	O(1)	O(1)	RO	✓*	✓*		✓*	1 – (1 – f) C
DPDP I	O(log n)	O(log n)	O(log n)	standard	✓	✓	✓	✓	1 – (1 – f) C
DPDP II	O(n ϱ log n)	O(log n)	O(log n)	standard	✓	✓	✓	✓	1 – (1 – f) Ω(log n)

Figure 6 - Comparison of PDP schemes: original PDP scheme, Scalable PDP, DPDP I authenticated skip lists, DPDP II scheme based on RSA trees. A star (*) indicates that a certain operation can be performed only a limited (pre-determined) number of times.

There is a price to use the dynamic provable data possession which is a performance change *"change from O(1) to O(log n) (or O(n ϱ log n)), for a file consisting of n blocks, while maintaining the same (or better, respectively) probability of misbehaviour detection"* (Erway, et al., 2009).

But in section 8.2 of the article, they run several experiments to demonstrate the performance of their schema. Here are two charts that demonstrate the performance difference between the PDP and DPDP scheme –

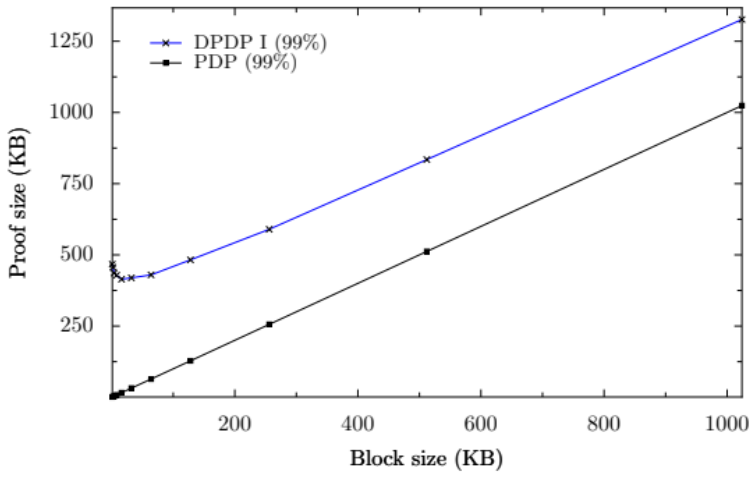


Figure 7 - Size of proofs of possession

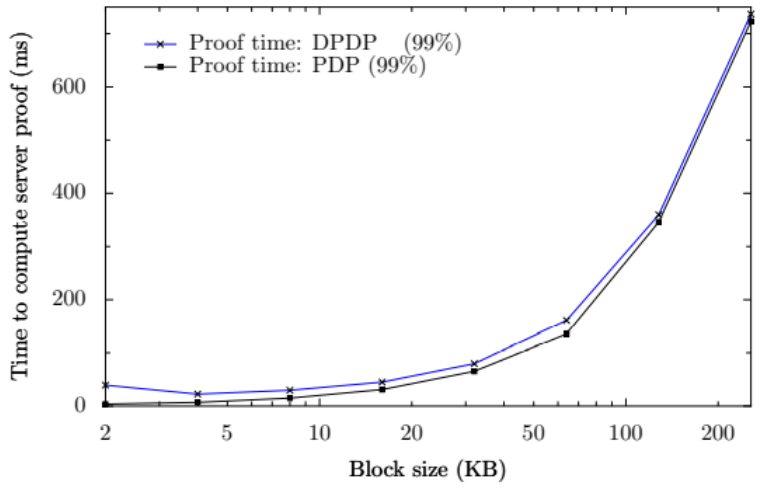


Figure 8 - Computation time required by the server in response to a challenge

Figure 7 - Size of proofs of possession Figure 7 - Size of proofs of possessionshows the size of the proof generated on a 1GB with a 99% probability of detecting misbehaviour. Figure 8 - Computation time required by the server in response to a challenge shows the computation time required by the server in response to a challenge for a 1GB file. There experiment go on to prove that *"a block size of 16KB is best for 99% confidence, resulting in a proof size of 415KB, and computational overhead of 30ms. They also show that the price of dynamism is a small amount of overhead compared to the existing PDP scheme"* (Erway, et al., 2009).

4.4. Provable Data Possession with Outsourced Data Transfer (Wang, et al., 2019)

Since (Erway, et al., 2009) there has not been much development in the PDP scheme it self rather than research done in niche areas adopting and enhancing the schema for specific needs.

(Wang, et al., 2019) published an article showing how they adapted the PDP scheme to handle data transfer between too clouds/ on premise store and *"guarantee the remote data integrity when the data are maintaining on the cloud servers and are transferring between two clouds, and secure deletion of transferred data on the original cloud."*.

They achieved this by creating a protocol where the new cloud provider can generate a proof to convince *"users that the data are securely and entirely transferred to its servers without any corruption and the original cloud has been erased all the data of users"* (Wang, et al., 2019)

4.5. Summary

Provable data possession is essential for being able to prove that the client's file is intact and has not been tampered with, without requiring the tenant to download the actual data.

Above is a detailed history of this schema starting with Guiseppe, et al., (2007). Every single version has its benefit and drawback but with each variation, the drawbacks are becoming less and less.

Guiseppe, et al., (2007) was a great starting point for the PDP schema and is still a great choice for verifying archived data that does not need to be modified. But in today's climate many companies and organisation store all their data on an untrusted server, so there is a need to verify dynamic data.

This is where Ateniese, et al., (2008) and Erway, et al., (2009) extends Guiseppe, et al., (2007) adding the ability to efficiently verify dynamic data. Ateniese, et al., (2008) adds the ability to be able to modify and delete without needing to run through the entire schema again. But Ateniese, et al., (2008) still has some drawbacks, that are discussed above. Erway, et al., (2009) goes a step further by also allowing block insertions, but they still suffer from small performance issues that they go on to justify.

Chapter 5: Proofs of Retrievability (POR)

Proofs of Retrievability (POR), is very similar to the PDP schema. PDP demonstrates to a client that a server possesses the client's file and it has not been modified or deleted. POR allows the client to run an efficient audit protocol where the server proves that the client's file can be retrieved. POR schemes also have the ability to retrieve and fix files that has small file corruption with the use of error-correcting codes.

5.1. PORs: Proofs of Retrievability for Large Files (Juels & Kaliski, 2007)

This schema is very similar to the provable data possession schema mentioned above. The primary difference is that the Proof of Retrievability (POR) schemas focuses on the means for the client to receive proof that their data is begin stored without corruption and with the ability to retrieve the entire file even if the file has 'small file corruptions'.

This schema much like the first iteration of the PDP schema focuses on static storage and is designed for archived data.

The schema is quite simple. It encrypts the file (F) and randomly adds check blocks which they have called 'sentinels'. And with the *"use of encryption here renders the sentinels indistinguishable from other file blocks"* (Juels & Kaliski, 2007). The client then challenges the storage provider on these sentinels. It does this by *"specifying the positions of a collection of sentinels"* (Juels & Kaliski, 2007) and then asking the storage provider to *"return the associated sentinel values"* (Juels & Kaliski, 2007). If the storage provider modified or deleted part of the file (F), there will be a high probability it will have also have suppressed several sentinels (Juels & Kaliski, 2007), and will be unlikely to respond with correct file blocks that correspond with the sentinels generated during the setup phase.

To protect against corruption, they *"also employ error-correcting codes"* (Juels & Kaliski, 2007). This is to reveal small file corruptions that could be missed between sentinels. This means that the sentinels are used to detect if a large portion of the file has been modified or corrupted, and it would be unlikely to be able to retrieve or repair the file. If small parts of the file are corrupted, likely, this will not be detected but with the use of the error-correcting codes, the file will be retrieved and repaired.

Figure 9 - POR setup process (Gilberg, 2014) demonstrates the setup process of this schema –

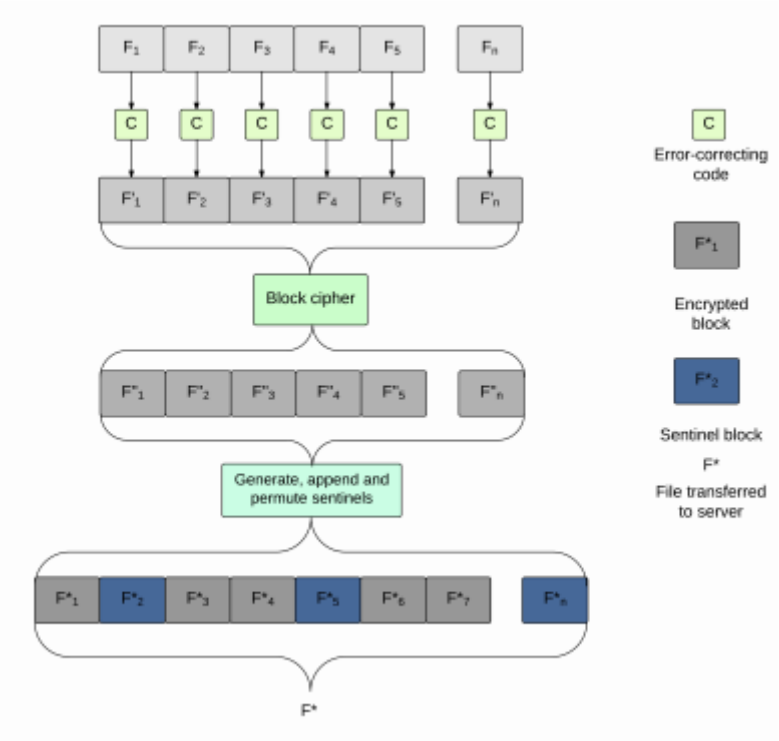


Figure 9 - POR setup process

The main drawback of this process *"is the pre-processing / encoding of F required prior to storage"* (Juels & Kaliski, 2007). The process of embedding sentinels and error-correcting codes imposes some computational overhead and cause larger storage requirements on the storage provider.

As you can see from the diagram above Figure 9 - POR setup process, a file is split into blocks (F1 … Fn) and error-correcting code is added to each block. The resulting blocks then go through a block cypher encrypting the file. The final step generates the sentinels that are applied to the encrypted file.

The above steps are all executed on the client before the file (F*) is transferred to the server.

The sentinels are a small fraction of the encoded file, typically 2% (Juels & Kaliski, 2007), but the error-correcting codes *"imposes the bulk of the storage overhead"* (Juels & Kaliski, 2007). And for larger files *"the associated expansion factor |F˜|/|F| can be fairly modest, e.g., 15%."* (Juels & Kaliski, 2007).

Here are two examples that the authors give for the different scenarios mentioned above –

1. *"Suppose that the prover, on receiving an encoded file F˜, corrupts three randomly selected bits, b1, b2, b3. These bits are unlikely to reside in sentinels, which constitute a small fraction of F˜. Thus, the verifier will probably not detect corruption through POR execution. Thanks to the error-correction present in F˜, however, the verifier can recover the original file F completely intact"* (Juels & Kaliski, 2007)
2. "Suppose conversely that the prover corrupts many blocks in F˜, e.g., 20% of the file. In this case (absent very heavy error-coding), the verifier is unlikely to be able to recover the original file F. On the other hand, every sentinel that the verifier requests in a POR will detect the corruption with probability about 1/5. By requesting hundreds of sentinels, the verifier can detect the corruption with overwhelming probability" (Juels & Kaliski, 2007)

The use of sentinels and error-correcting codes improve the error-resiliency of their system. Unfortunately, this means that it does not support updates, without simply replacing the entire file with a new file. It also means the number of queries the clients can make is fixed, which puts a restriction on the lifetime of the scheme (Erway, et al., 2009).

5.2. Compact Proofs of Retrievability (Shacham & Waters, 2008)

Shacham and Waters (2008) have improved on Juels & Kaliski, (2007) schema called Compact Proofs of Retrievability (Shacham & Waters, 2008), but their solution is also static.

In their paper, they explain and demonstrate two versions of their schema.

1. The first one is *"built from BLS signatures and secure in the random oracle model"* (Shacham & Waters, 2008), which has the shortest query and response of any POR system.
2. And there second schema *"which builds elegantly on pseudorandom functions (PRFs) and is secure in the standard model"* (Shacham & Waters, 2008), which has the shortest response of any POR system.

Both are based on using a homomorphic authenticator for file block. Which essentially means that *"block integrity values that can be efficiently aggregated to reduce bandwidth in a POR protocol"* (Bowers, et al., 2009)**.** Juels & Kaliski, (2007) scheme use MAC-Based

message authentication, which according to Bowers, et al., (2009) would increase the size of the response *"If each authenticator is λ bits long, as required in the Juels-Kaliski model, then the response is λ 2 · (s + 1) bits, where the ratio of file block to authenticator length is s : 1.9"* (Bowers, et al., 2009)

The use of homomorphic authenticators rather than MAC-based message improves the response length. Homomorphic authenticators are explained in more detail by Krohn, et al., (2004), who stated: *"It is fast to compute, efficiently verified using probabilistic batch verification, and has provable security under the discrete-log assumption".*

The main advantages of this schema over Juels & Kaliski, (2007) is the smaller response length and unlike the Juels and Kaliski (2007) scheme, you are not limited in the number of verification you can do.

However, Shacham and Waters (2008) still have the same drawback and that is that it only works for static file archival and you are not able to update or modify the file without removing the original file and re-uploading.

5.3. Proofs of Retrievability: Theory and Implementation (Bowers, et al., 2009)

This scheme improves on the work done by Juels & Kaliski (2007) and Shacham & Waters (2008).

Bowers, et al., (2009) schema is a variant on the Juels & Kaliski (2007) POR scheme, and they used it as a starting point. Bowers, et al., (2009) have improved on two key part of a POR system and that is, allowing for higher acceptance of error rate, while still being able to retrieve the original file. They have also managed to lower the data overhead on the uploaded file.

"Designing the new variant on JK are to tolerate a larger level of errors than in the original JK scheme, reduce the storage overhead on the server" (Bowers, et al., 2009)**.**

The error-correcting method that Bowers, et al., (2009) is different from Juels & Kaliski (2007) and Shacham & Waters (2008), Bowers, et al., (2009) use an inner and outer error-correcting code which allows a higher error tolerance rate.

Bowers, et al., (2009) describe the inner and outer code as *"The two codes play complementary roles, but operate in distinct ways and at different protocol layers"* and that

the inner code is computed on the fly by the server and which *"creates no storage overhead"* (Bowers, et al., 2009) but does *"imposes a computational burden on the server when it responds to client challenges"* (Bowers, et al., 2009). This is because the server must retrieve the selected blocks from the challenge and apply the inner code each time.

The outer code has a similar effect to the error-correcting code in Juels & Kaliski (2007) and Shacham & Waters (2008), schema where it has little effect on the servers computational power, but does increase the stored file size, therefore, the outer error-correcting code is embedded with the file.

5.4. Improved Proofs Of Retrievability And Replication For Data Availability In Cloud Storage (Guo, et al., 2020)

Similarly to the progression of the PDP schema, there has not been much development in the underling schema since 2009 (Bowers, et al., 2009), but there has been develpement in adopting it for specific needs.

(Guo, et al., 2020) has modiefid the POR schema to not only focusing on data integrity, but data availability when there is a server failure. They achieved this by utilising the coulds ability to replicate the data for redundance. They then adapted the POR schema to *"ensure that if some of replicas are corrupted, the file can still be restored by means of the healthy replicas"* (Guo, et al., 2020). In order to achieve this then needed to prove that "multiple replicas of the file are indeed stored" (Guo, et al., 2020).

They started by identifying existing solutions for this problem such as,

- Multiple-Replica Provable Data Possession (Curtmola, et al., 2008)
- Transparent, Distributed, and Replicated Dynamic Provable Data Possession (Etemad & Kupcu, 2013)
- Provable Multicopy Dynamic Data Possession in Cloud Computing Systems (Barsoum & Hasan, 2014)
- MuR-DPA: Top-Down Levelled Multi-Replica Merkle Hash Tree Based Secure Public Auditing for Dynamic Big Data Storage on Cloud (Lui, et al., 2014)

After they reviewed the above they came to the conclusion that all solved separate issues but still wasn't exactly what they where looking for. They then discussed a solution called

'Mirror' (Armknecht, et al., 2016), which does what they wanted but they identified numourse security flaws with the implementation. Mirror allows *"users process and upload their stored file only once regardless of the number of replicas required. The generation of replicas is completely done by the provider to save expensive bandwidth resource; thus, there is no chance for users to misbehave. On the other hand, Mirror exploits cryptographic puzzles to defend against a dishonest provider who does not store the replicas correctly and tries to compute them on the fly when needed. Unfortunately, we find that Mirror is vulnerable to some potential attacks that might be mounted by a dishonest provider. This imposes new security risks on users and is a drawback that should be further conquered."* (Guo, et al., 2020)

They then continue the article by demonstrating the current issues with Mirror, and demonstrated solutions to the problems.

The concluded the article by fixing the issues with the Mirror implementation by adding a new *"parameter S should be chosen randomly in each challenge that indicates which sectors of each involved replica block will be checked in the challenge. In addition, we developed a modification of the authentication tag, which not only enables users to perform the verification efficiently but also can prevent from substitution attack and forgery attack. Finally, we presented an IPOR2 scheme, which achieves a high security and retains the advantages of Mirror. Our evaluation results showed that the IPOR2 scheme performs comparable to Mirror while providing a strong security guarantee"* (Guo, et al., 2020).

5.5. Summary

Proofs of retrievability is essential to proving that a client can still retrieve the entire file without corruption.

In the section analysed a number for different POR approaches starting with Juels & Kaliski (2007). Much like the start of the PDP schema they too focused on static data, meaning that this schema does not support updates. Another drawback of this schema is the number of queries the clients can make is fixed, which puts a restriction on the lifetime of the scheme.

Next came a paper title Compact Proofs of Retrievability Shacham & Waters (2008), which improved the Juels & Kaliski (2007) schema but they too only supported static data. Shacham & Waters (2008) improved Juels & Kaliski (2007) schema in two ways, first, they

made the response smaller improving the bandwidth usage, and their schema is not limited in the number of verification you can do.

Bowers, et al., (2009) schema which was based on the works of Juels & Kaliski (2007) and Shacham & Waters (2008) improved two key parts of the POR system, allowing for higher acceptance of error rate, while still being able to retrieve the original file. They have also managed to lower the data overhead on the uploaded file. But the POR schemas mentioned above all share the same drawback, and that is they only support static files.

Chapter 6: Proposed Conceptual model

This section will be discussing a conceptual model based on the research done in the above sections. To create this conceptual model, identifying the strength and weaknesses of both approaches must be done first. Table 2 - Provable data possession (strengths and weaknesses)and Table 3 - Proofs of retrievability (strengths and weaknesses) identifies the strengths and weaknesses of each approach.

6.1. Provable data possession (strengths and weaknesses)

Strength	Weaknesses
Proves that a file is intact and has not been tampered with.	You have to decide between flexibility and performance. Currently adding the ability of block insertions decreases the performance.
Doesn't require downloading the entire file.	Does not guarantee that the client can retrieve the file.
In the later approaches, the schema is more flexible allowing – appending, modifying, inserting, deleting entire blocks, without needing to run through the entire process again.	
Good use of bandwidth.	

Table 2 - Provable data possession (strengths and weaknesses)

6.2. Proofs of retrievability (strengths and weaknesses)

Strength	Weaknesses
Proves that the file is retrievable (without corruption).	Not flexible. Currently only works with static data.
Fix files with small file corruptions.	The number of queries the clients can make is fixed, which puts a restriction on the lifetime of the scheme.
Good use of bandwidth.	Data expansion due to additional sentinel blocks.
Doesn't require downloading the entire file.	

Table 3 - Proofs of retrievability (strengths and weaknesses)

Both models are very similar to each other –

- Both rely on metadata being stored on the client
- Both pre-processes the file on the client
- Both attempts to limit the size of bandwidth used
- Both attempts decrease latency and time taking to perform the checks

The differences lie within the goal of each approach, which are –

- PDP - proves that a file is intact and has not been tampered with
- POR - proves that the file is retrievable (without corruption).

The conceptual model proposed in this thesis is a combination of both models with the end goal of the model to be able to prove that the file is intact and retrievable.

This model is based on the PDP model created by Erway, et al., (2009). The model processed will be based on Erway, et al., (2009) PDP due to it being the most advanced model and has the fewest limitations of the three discussed in this research.

The decision was made to base the model on the PDP model over the POR model because it is the more complicated model and, would be simpler to implement POR model into the PDP model rather PDP into the POR model.

There are two key aspects to the POR model that differs from the PDP model –

1. Blocks which they have called 'sentinels'
2. Error-correcting code

The first is the use of check blocks 'sentinels', these are blocks of data used to challenge the server at a later date. Sentinels are indistinguishable from other file blocks and the server will be asked to return specific file blocks to prove that the file is retrievable.

Then there are error-correcting codes, these are created to protect against corruption Juels & Kaliski (2007) and are used to reveal small file corruptions that could be missed between sentinels.

This means that the sentinels are used to detect if a large portion of the file has been modified or corrupted, and it would be unlikely to be able to retrieve or repair the file. If small parts of the file are corrupted, likely, this will not be detected but with the use of the error-correcting codes, the file will be retrieved and repaired.

If both of these aspects can be merged into the PDP model then the model would have the benefits of both the PDP and POR model.

Here is a diagram for the conceptual model. Much like the other PDP and POR schema, there are two diagrams, first to show the pre-processing and upload of the file and the other to query the server for the proof (possession and retrievability).

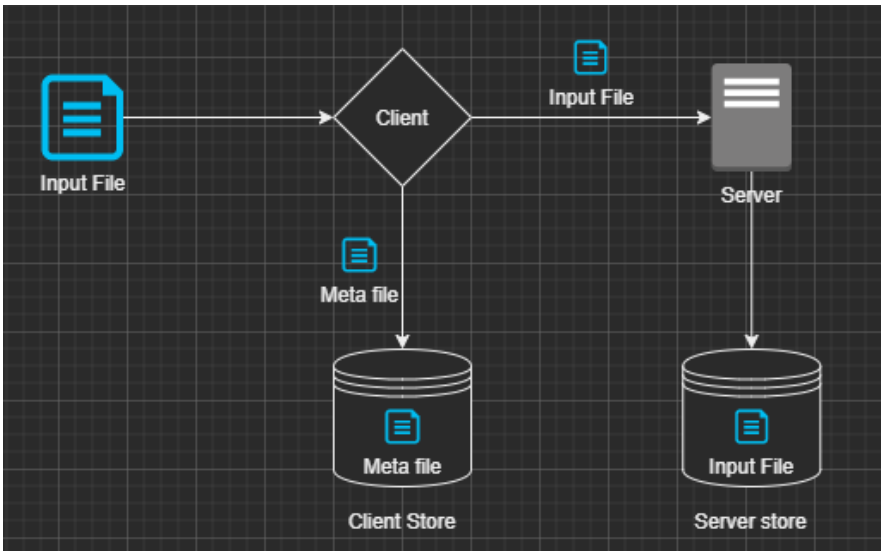


Figure 10 - File upload

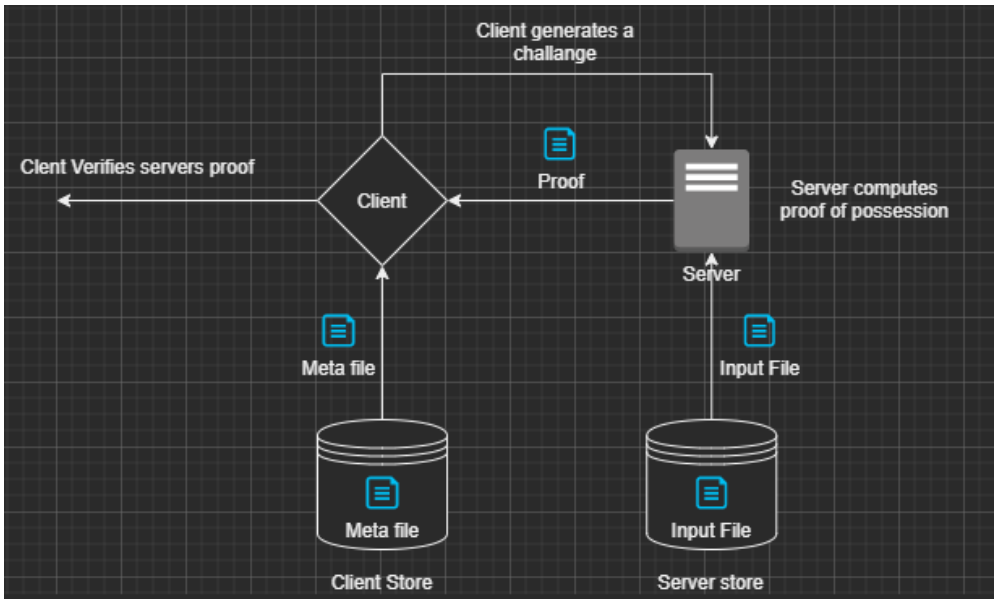


Figure 11 - Retrieve proof

As you can see the diagram are remarkably similar to the PDP diagrams shown earlier in this thesis. The main difference between the two is the pre-processing of the file on the client to embed more information into the file and to store slightly more meta data to be able to benefit from both approaches.

The issues this research foresees is that more research will be needed to make the POR model compatible with dynamic data before the model can be implemented. At this current point of time attempting to merge both approaches would limit the model to static data only. A paper by Curtmola, et al., (2008) added the error-correcting codes to the PDP schema, but their research was based on the original PDP schema which was only compatible with static data. The other issue is around server overhead with both the PDP and POR process adding more data to the file it would increase the server overhead and bandwidth.

Since writing this thesis a new paper was published called *'Dynamic proofs of retrievability with low server storage'* (Anthoine, et al., 2021). Due to the date the article was published a critical analysis of it has not been proformmed, but from the out set it's seams to have fixed one of the main issues identified above, *'the POR model compatible with dynamic data'.* Further analysis is needed to investigate this new POR approach and identify the strenghs and weaknesses of their approach. If they have fixed the dynamic data issue with the POR schema, I'd recommend using the POR schema and combining it with the PDP schema the above conceptual model is based on.

Chapter 7: Conclusion

Storage-outsourcing and resource sharing networks have gained popularity over the last few years. This is due to several factors such as –

- Scalability and Flexibility
  "Cloud-based services are ideal for businesses with growing or fluctuating bandwidth demands. If your needs increase, it is easy to scale up your cloud capacity, drawing on the service's remote servers." (Slack, 2019). This also works in reverse if bandwidth or computational needs go down the service automatically scale, saving on costs.
- Low-Cost and Capital-Expenditure
  "Establishing and running a data centre is expensive. You need to purchase the right equipment and hire technicians to install and manage the centre. When you shift to cloud computing, you will only pay for the services procured." (Sumner, 2017). With many cloud providers, you only pay for what you use.
- Recovery and Data recovery
  Many cloud providers offer data recovery as standard. Meaning it "Eliminate the need to replicate your production system in full at a secondary company managed data centre" (Singh, 2016)

The lower cost and the ease of scaling on demand which would become extremely expensive to achieve without cloud providers such as 'Amazon web service' and 'Microsoft Azure'. Many new and small companies now are looking to cloud providers before looking in-house because of the cost-saving ability and the ease of management, and many large companies such as 'Netflix', 'Pinterest' and 'Symantec' (Stokel-Walker, 2017).

The sudden increase in shift also came with a lot of trust issues, many large companies simply do not want to hand over all their precious data to an untrusted third party. The problem of *"efficiently proving integrity of data stored at untrusted servers has received increased attention"* (Erway, et al., 2009)**.**

In the chapters *`Chapter 4: Provable data possession`* and *`Chapter 5: Proofs of Retrievability (POR)`* a chronological analysis was done on the progression of both schemas. In the papers that present the PDP/ POR schemes, they usually criticise each other. In

Bowers, et al., (2009) they noted that the *"PDP demonstrates to a client that a server possesses a file F (in an informal sense), but is weaker than a POR in that it does not guarantee that the client can retrieve the file".* But on the other side Ateniese, et al. (2008) and Erway, et al., (2009) both criticize the POR schema saying that the PDP ***"****is more efficient than POR as it requires no bulk encryption of outsourced data and no data expansion due to additional sentinel blocks***"** (Ateniese, et al., 2008) and that the POR systems "prevent any efficient extension to support updates, beyond simply replacing F with a new file F '. Furthermore, the number of queries a client can perform is limited and fixed a priori" (Erway, et al., 2009)**.**

PDP and POR schemas do have different responsibilities. PDP main aim is to ensure that the client file is intact and has not been tampered, whereas POR main aim is to guarantee that the client can retrieve the file even with small file corruption. Both have real-world usage and is critical for today's data-centric world.

Merging both of these schema could have huge benafits. In chapter *'Chapter 6: Proposed Conceptual model'* propsas a conceptual model based on the analysis of previous papers. A previous paper called 'Robust remote data checking' (Curtmola, et al., 2008) proved that this is a possibility, issue with there research is that it was developed quite early in the development of both schema which means that it doesn't support dynamic data, as at the time neither of the schema did.

More research is needed to make the POR model compatible with dynamic data before the model can be implemented. At this current point of time attempting to merge both approaches would limit the model to static data only, which would have little benefit over the previous research.

Once POR does support dynamic data attepting to merge both schemas should be successful, but this research does forsea an issue with server overhead with both the PDP and POR process adding more data to the file it would increase the server overhead and bandwidth, so further research will be needed to make the process more efficient.

7.1. Limitations

This research focusses on understanding and critically analysing the current state of 'Provable data possession' and 'proofs of retrievability', and using the information gathered to create a conceptual model. They are a few limitations of this thesis though.

The entire thesis is based on secondary data, meaning someone can enhance this research conducting their own experiments to confirm the secondary data is accurate.

A limitation on this research is that the scope of this research was focused around two specific approaches. The research done in this thesis can be used to compare against other technologies/ approaches trying to achieve the same result.

A new paper was published called *'Dynamic proofs of retrievability with low server storage',* due to the date the article was published a critical analysis of it has not been proformmed, but from the out set it's seams to have fixed one of the main issues identified above. Furfther research is needed to investigate the contributions of this paper, and whether it's solves one of the main issues identified in this thesis.

The final limitation is the model created is only conceptional, meaning further research can be done to implement and test the model. Once implemented the data used within this research can be used to evaluate its potential.

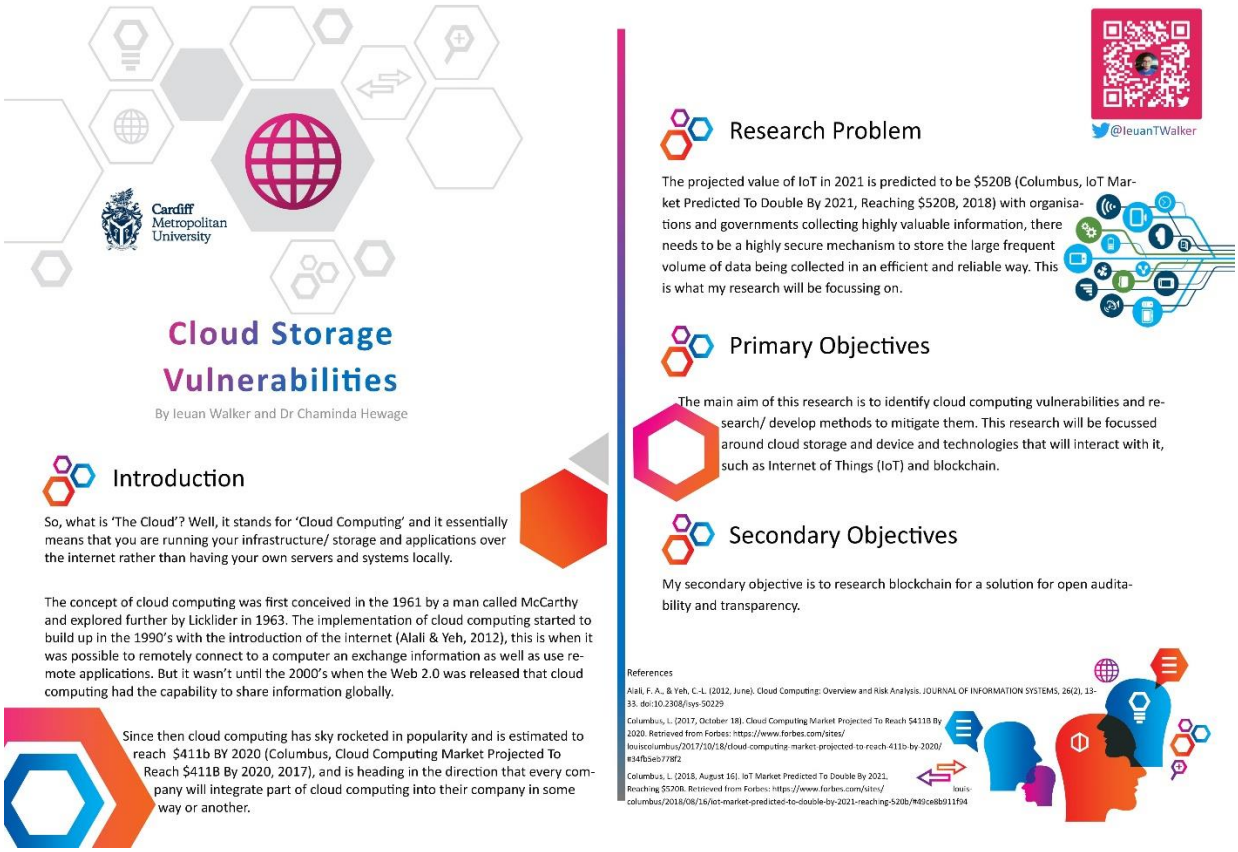
7.2. Future work

The future work that can be done based on this work is the implementation of the conceptual model. Using the model, a comparison can then be done on performance, security, latency and efficiency of the model based on the PDP and POR schema.

7.3. Outputs

CRESTCon

CRESTCon is an International Technical Cyber Security Industry Conference and Exhibition. The conference was held on 14th March 2019 at Royal College of Physicians, London. I was selected to present my research in poster form to other people in the Cyber Security area.



Speaking of science poster

Speaking of science is a student-led conference which is an excellent opportunity to network with peers. It allows you to learn what others in your field are researching, what results they are getting and the methodologies they use to get them. I was

selected to present my research in poster form.

References

Alali, F. A. & Yeh, C.-L., 2012. Cloud Computing: Overview and Risk Analysis. *JOURNAL OF INFORMATION SYSTEMS,* June, 26(2), pp. 13-33.

Albor, V. M. F., Pena, A. T. F., Diaz, R. G. & Silva, J. J. S., 2015. *Platform as a Service intergration for scientific computing using dirac.* s.l.:s.n.

Amazon, 2018. *SaaS on AWS.* [Online]
Available at: https://aws.amazon.com/partners/saas-on-aws/
[Accessed 13 November 2018].

Anthoine, G. et al., 2021. *Dynamic proofs of retrievability with low server storage.* s.l.:s.n.

Armknecht, F., Barman, L., Bohli, J.-M. & Karame, G. O., 2016. *Mirror: Enabling Proofs of Data Replication and Retrievability in the Cloud.* Austin, TX, usenix.

Ateniese, G., Pietro, R. D., Mancini, L. V. & Tsudik, G., 2008. *Scalable and Efficient Provable Data Possession.* Istanbul, Turkey, ACM, pp. 9:1-9:10.

Balasubramanian, K., 2014. *Variants of RSA and their cryptanalysis.* Sivakasi, India, IEEE, pp. 145-149.

Barsoum, A. F. & Hasan, A. M., 2014. Provable Multicopy Dynamic Data Possession in Cloud Computing Systems. *IEEE Transactions on Information Forensics and Security,* 18 December, 10(3), pp. 485 - 497.

Bhardwaj, S., Jain, L. & Jain, S., 2010. Cloud Computing: A study of Infrastructure as a Service (IaaS). *International Journal of Engineering and Information Technology,* 2(1), pp. 60-63.

Bowers, K. D., Juels, A. & Oprea, A., 2009. *Proofs of retrievability: theory and implementation.* Chicago, Illinois, USA, ACM, pp. 43-54.

Carroll, A., 2019. *WHY SHOULD YOU OUTSOURCE THE DATA CENTER?.* [Online]
Available at: https://lifelinedatacenters.com/data-center/benefits-of-outsourcing-data-centers/
[Accessed 20 December 2019].

Cloud Security Alliance, 2018a. *History.* [Online]
Available at: https://cloudsecurityalliance.org/history/
[Accessed 12 December 2018].

Cloud Security Alliance, 2018b. *Executive Members.* [Online]
Available at: https://cloudsecurityalliance.org/membership/corporate/
[Accessed 16 December 2018].

Cloud Security Alliance, 2018c. *About.* [Online]
Available at: https://cloudsecurityalliance.org/about/
[Accessed 12 December 2018].

Cobb, M., n.d. *Advanced Encryption Standard (AES).* [Online]
Available at: https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard
[Accessed 01 April 2021].

Cobb, M., n.d. *RSA algorithm (Rivest-Shamir-Adleman).* [Online]
Available at: https://searchsecurity.techtarget.com/definition/RSA
[Accessed 1 April 2021].

Curtmola, R., Khan, O. & Burns, R., 2008. *Robust remote data checking.* s.l., StorageSS, pp. 63-68.

Curtmola, R., Khan, O., Burns, R. & Ateniese, G., 2008. *Multiple-Replica Provable Data Possession.* Beijing, China, IEEE, pp. 411-420.

Dennis, B., 2010. Don't cloud your compliance data. *Control Engineering,* January, 57(1), p. 23.

Deswarte, Y., Quisquater, J.-J. & Sadane, A., 2004. Remote Integrity Checking. In: S. Jajodia & L. Strous , eds. *Integrity and Internal Control in Information Systems.* Boston, MA: Springer, pp. 1-11.

Devanbu, P., Gertz, M., Martel, C. & Stubblebine, S. G., 2002. *Authentic Third-Party Data Publication.* Boston, MA, Springer, pp. 101-112.

Dodis, Y., Vadhan, S. & Wichs, D., 2009. *Proofs of Retrievability via Hardness Amplification.* Berlin, Heidelberg, Springer, pp. 109-127.

Erway, C., Küpçü, A., Papamanthou, C. & Tamassia, R., 2009. *Dynamic provable data possession.* Chicago, Illinois, USA, ACM, pp. 213-222.

Etemad, M. & Kupcu, A., 2013. *Transparent, Distributed, and Replicated Dynamic Provable Data Possession.* Berlin Heidelberg, Springer-Verlag, pp. 1-18.

European Union Agency for Cybersecurity (ENISA), 2019. *About ENISA.* [Online]
Available at: https://www.enisa.europa.eu/about-enisa
[Accessed 13 November 2019].

Filho, D. L. G. & Barreto, P. S. I. M., 2006. Demonstrating data possession and uncheatable data transfer. *IACR Cryptology,* Volume 2006, p. 150.

Forcepoint, n.d. *Data encryption.* [Online]
Available at: https://www.forcepoint.com/cyber-edu/data-encryption
[Accessed 05 April 2020].

Gilberg, O. R., 2014. *Cloud Security without Trust.* s.l.:Norwegian University of Science and Technology Department of Telematics.

Golle, p., Staddon, J. & Waters, B., 2004. *Secure Conjunctive Keyword Search over Encrypted Data.* Berlin, Heidelberg, Springer, pp. 31-45.

Guiseppe, A. et al., 2007. *Provable Data Possession at Untrusted Stores.* Alexandria, Virginia, USA, ACM, pp. 598-609.

Guo, W. et al., 2020. Improved Proofs Of Retrievability And Replication For Data Availability In Cloud Storage. *The Computer Journal,* 63(8), p. 1216–1230.

Interoute, 2019. *What is private cloud?.* [Online]
Available at: https://www.interoute.com/what-private-cloud
[Accessed 11 March 2019].

Joshi, G., 2012. *The different types of cloud and their relevance.* [Online]
Available at: https://cloudtweaks.com/2012/04/types-of-cloud-and-their-relevance/
[Accessed 11 March 2019].

Juels, A. & Kaliski, B., 2007. PORs: Proofs of Retrievability for Large Files. *Proceedings of the 14th ACM Conference on Computer and Communications Security,* pp. 584-597.

Kallahalla, M. et al., 2003. *Plutus: Scalable secure file sharing on untrusted storage.* San Francisco, CA, USA, USENIX Association, pp. 29-42.

Ko, D. & Boutelle, J., 2018. *Multicloud Storage.* 2nd HPE Special Edition ed. New York: John Wiley & Sons, Inc..

Krohn, M. N., Freedman, M. J. & Mazieres, D., 2004. *On-the-Fly Verification of Rateless Erasure Codes.* Berkeley, CA, USA, IEEE.

Kubiatowicz, J. et al., 2000. *OceanStore: an architecture for global-scale persistent storage.* s.l., ACM SIGPLAN Notices.

Lui, C. et al., 2014 . MuR-DPA: Top-Down Levelled Multi-Replica Merkle Hash Tree Based Secure Public Auditing for Dynamic Big Data Storage on Cloud. *IEEE Transactions on Computers,* 26 November , 64(9), pp. 2609 - 2622.

Lui, F. et al., 2011. *NIST Cloud Computing Reference Architecture.* s.l.:s.n.

Maheshwari, U., Vingralek, R. & Sharpiro, W., 2000. *How to Build a Trusted Database System on Untrusted Storage.* Santa Clara, CA, Proceedings of the 4th conference on Symposium on Operating System Design & Implementation.

McLeod, S., 2019. *What's the difference between qualitative and quantitative research?.* [Online]
Available at: https://www.simplypsychology.org/qualitative-quantitative.html
[Accessed 2020 March 15].

Mell, P. & Grance, T., 2011. *Recommendations of the National Institute of Standards and Technology.* [Online]
Available at: https://csrc.nist.gov/publications/detail/sp/800-145/final
[Accessed 27 November 2018].

Mell, P. & Grance, T., 2011. *Recommendations of the National Institute of Standards and Technology.* [Online]

Available at: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
[Accessed 08 February 2019].

Microsoft, 2018. *What is PaaS? Platform as a service.* [Online]
Available at: https://azure.microsoft.com/en-gb/overview/what-is-paas/
[Accessed 13 November 2018].

Microsoft, 2019. *What is a hybrid cloud?.* [Online]
Available at: https://azure.microsoft.com/en-gb/overview/what-is-hybrid-cloud-computing/
[Accessed 11 March 2019].

National Institute of Standards and Technology, 2011. *Final Version of NIST Cloud Computing Definition Published.* [Online]
Available at: https://www.nist.gov/news-events/news/2011/10/final-version-nist-cloud-computing-definition-published
[Accessed 27 November 2018].

Quisquater, J.-J. & Fran¸cois-Xavier, S., 2005. *Exhaustive Key Search of the DES: Updates and Refinements.* s.l.:s.n.

Saunders, M., Lewis, P. & Thornhill, A., 2019. *Research methods for business students.* 8th ed. Harlow, United Kingdom: Pearson Education.

Shacham, H. & Waters, B., 2008. Compact Proofs of Retrievability. *Journal of Cryptology,* 01 July, 26(3), pp. 442-483.

Singh, J., 2016. *20 benefits of moving backup and disaster recovery to the cloud.* [Online]
Available at: https://www.itproportal.com/2016/03/31/20-benefits-moving-backup-and-disaster-recovery-to-cloud/
[Accessed 14 December 2019].

Slack, C., 2019. *10 Reasons that so many businesses are moving to the cloud.* [Online]
Available at: https://www.bluesilvershift.com/why-are-so-many-businesses-moving-to-the-cloud-2/
[Accessed 10 December 2019].

Stokel-Walker, C., 2017. *Case studies in cloud migration: Netflix, Pinterest, and Symantec.* [Online]
Available at: https://increment.com/cloud/case-studies-in-cloud-migration/
[Accessed 11 December 2019].

Sumner, J., 2017. *7 reasons why your business needs to move to the cloud.* [Online]
Available at: https://www.growthbusiness.co.uk/7-reasons-business-needs-move-cloud-2552054/
[Accessed 13 December 2019].

Techopedia, 2018. *Cloud Security Alliance (CSA).* [Online]
Available at: https://www.techopedia.com/definition/26532/cloud-security-alliance-csa
[Accessed 16 December 2018].

The university of Warwick, n.d. *Inductive or deductive approaches.* [Online]
Available at: https://warwick.ac.uk/fac/soc/ces/research/current/socialtheory/maps/when/
[Accessed 20 June 2020].

Townsend, P., 2019. *RSA VS AES ENCRYPTION.* [Online]
Available at: https://info.townsendsecurity.com/rsa-vs-aes-encryption-a-primer#:~:text=You%20can%20combine%20RSA%20encryption,protecting%20it%20with%20RSA%20encryption.&text=AES%20is%20not%20the%20only%20symmetric%20encryption%20method.
[Accessed 1 April 2021].

Wang, H. et al., 2019. *Provable Data Possession with Outsourced Data Transfer.* s.l., IEEE, pp. 1-1.