University Institute of Lisbon

Department of Information Science and Technology

A Blockchain-based Information Security Solution for a Distributed Multi-Cloud Secure Storage System (DMCS^3)

João Paulo Serafim Lobato

A Dissertation presented in partial fulfillment of the Requirements for the Degree of

Master in Computer Science and Business Management

Supervisor

Prof. Dr. Carlos José Corredoura Serrão, Assistant Professor

ISCTE-IUL

October 2018

"Too often we underestimate the power of a touch, a smile, a kind word, a listening ear, an honest compliment, or the smallest act of caring, all of which have the potential to turn a life around."

Leo Buscaglia

Resumo

Com o aumento dos dados gerados a nível mundial impulsionado pelo rápido crescimento da Internet das coisas e dos aparelhos conectados, torna-se de extrema importância guardar e manter guardado, fora do alcance de pessoas indesejadas, tudo o que nos diz respeito e/ou que nos possa comprometer, mesmo que de forma inadvertida.

Nesta senda da segurança da informação e proteção de dados numa era da Internet das coisas cada vez mais conectada, surgiu a oportunidade de desenvolver, embora que apenas conceptualmente, uma solução que pretende atender aos requisitos cada vez mais exigentes das pessoas e das empresas, no que à segurança dos seus dados privados concerne.

É, portanto, apresentado neste trabalho um desenho conceptual de uma solução totalmente distribuída e descentralizada, passível de ser usada em qualquer ambiente conectado e com capacidade de processamento, com recurso a armazenamento distribuído num sistema multicloud e com recurso a tecnologia blockchain e sistemas peer-to-peer para autenticação tanto de utilizadores como de ficheiros partilhados. Serão também explicados detalhadamente todos os processos, desde a autenticação de um utilizador e de um dispositivo ao armazenamento efectivo de fragmentos de dados, ou ficheiros, em diversos serviços cloud e sua partilha com terceiros.

A sua capacidade teórica de resistir a ciber ataques, bem como possíveis melhorias nas áreas de segurança, rastreabilidade e auditabilidade serão aferidas em comparação com soluções actualmente existentes, e a sua disponibilidade teórica será também alvo de estudo, de forma a que um sistema similar possa ser futuramente alvo de estudo mais aprofundado e a sua viabilidade possa ser aferida num ambiente de testes propício a melhorias.

Palavras-chave: Segurança de Informação, Blockchain, Distributed Ledgers, Criptografia, Sistemas Distribuídos, Computação em Núvem.

Abstract

With the increase of data generated worldwide, driven by the rapid growth of the Internet of things and interconnected devices, it is extremely important to keep data and everything that concerns us or may compromise us in some way, even if inadvertently, away from undesired individuals.

While in pursuit of information security and data protection in an Internet of things driven, and even more connected society than before, the opportunity has arisen to develop, albeit only conceptually, a solution that intends to meet the increasingly challenging requirements people and companies demand nowadays, and to whom the security of their private data concerns.

It is therefore presented in this work a conceptual design of a totally distributed and decentralized solution, that can be used in any connected environment with processing capabilities, using distributed storage in a multicloud system and implementing blockchain technology and peer-to-peer authentication systems for both users, devices and shared files. All the processes will also be explained in detail, from the user and device authentication, to the effective storage of data fragments, or files, in various cloud services and its sharing capabilities with authorized third parties.

Its theoretical ability to resist cyber attacks as well as possible improvements in the areas of security, traceability and auditability will be benchmarked and compared against existing solutions, and its theoretical service availability will also be the subject of study so that a similar system can be further targeted for improvements and its feasibility can be measured in a suitable test environment in order to improve the presented solution.

Keywords: Information Security, Blockchain, Distributed Ledgers, Cryptography, Distributed Systems, Cloud Computing.

Acknowledgements

Due to his immense knowledge in the security field, his helpful insights and precious support on helping me achieving this degree and develop this work, I would like to publicly acknowledge my supervisor, Professor Carlos Serrão, and thanking him for all his support and dedication.

A very special thank you to my closest friends, both from the Military School and from the Naval Academy, for helping me thrive in life, for supporting me both in good and bad times, and for never letting me down. They are, without a doubt, the family I proudly chose.

My godfather Luís, my godmother Zulmira and my cousins Luís and Sofia Pereira, to whom I publicly acknowledge their support, kindness, dedication and friendship throughout my life. Both to me, and my family. They are an unquestionable example to follow and a chandelier that guide my path to the future.

To my brother Pedro Lobato, to Hugo, Marta and Zarita, who were not always with me but joined gradually with time, a big thank you for all your support and for putting up with me. I am not always correct or in a good mood, but nevertheless you are always around and, despite our differences of opinion and the way sometimes we see things differently, it makes me quietly think and reckon that family is what we should value the most in our lives.

A very special word of appreciation and gratitude to my sweet mother Fernanda, an angel on Earth, and my grandparents Antonieta and Dionísio who, despite everything that happened in our lives, despite all the difficulties they've been through, they have never stopped supporting me and were always present when I needed them the most. My life would be incomplete without them.

For last, the people I've spent most of my life with and to whom I owe everything: My father, to whom will never be enough to thank for everything he has done for me and all the sacrifices he made throughout his life in order to provide me the best he could, to my grandmother Eduarda, and to my grandfather Augusto, my role model in life, my inspiration in hard times and the most correct person I will ever meet in my life.

To all of you, a simple thank you wont be enough. Ever!

Contents

List of Figures

Abbreviations

DMCS³	**D**istributed **M**ulti-**C**loud **S**ecure **S**torage **S**ystem (see page i)
PII	**P**ersonally **I**dentifiable **I**nformation (see page 1)
IoT	**I**nternet **o**f **T**hings (see page 1)
BDLT	**B**lockchain and **D**istributed **L**edger **T**echnologies (see page 2)
IaaS	**I**nfrastructure **a**s **a** **S**ervice (see page 6)
VMs	**V**irtual **M**achine**s** (see page 6)
PaaS	**P**latform **a**s **a** **S**ervice (see page 6)
SaaS	**S**oftware **a**s **a** **S**ervice (see page 6)
NIST	**N**ational **I**nstitute of **S**tandards and **T**echnology (see page 6)
CSPs	**C**loud **S**torage **P**rovider**s** (see page 8)
DDoS	**D**istributed **D**enial **o**f **S**ervice (see page 9)
AWS	**A**mazon **W**eb **S**ervices (see page 9)
HAIL	**H**igh **A**vailability **I**ntegrity **L**ayer (see page 11)
IS	**I**nformation **S**ecurity (see page 10)
MCDB	**M**ulti-**C**loud **D**atabase **M**odel (see page 12)
USA	**U**nited **S**tates of **A**merica (see page 12)
NSA	**N**ational **S**ecurity **A**gency (see page 12)
TTP	**T**rusted **T**hird **P**arty (see page 13)
TCSEC	**T**rusted **C**omputer **S**ystem **E**valuation **C**riteria (see page 13)
CIA	**C**onfidentiality, **I**ntegrity and **A**vailability (see page 15)
CISO	**C**hief **I**nformation **S**ecurity **O**fficer (see page 16)
MPKCs	**M**ultivariate **P**ublic-**K**ey **C**ryptosystem**s** (see page 24)
ECC	**E**lliptic **C**urve **C**ryptography (see page 24)
P2P	**P**eer-to-**P**eer (see page 27)

UI **U**ser **I**nterface (see page 31)

PKIs **P**ublic **K**ey **I**nfrastructure**s** (see page 37)

ID **Id**entity (see page 48)

S3 **S**imple **S**torage **S**ervice (see page 64)

SLA **S**ervice **L**evel **A**greement (see page 65)

APIs **A**pplication **P**rogramming **I**nterface**s** (see page 67)

HD **H**ard **D**rive (see page 67)

SIEM **S**ecurity **I**nformation and **E**vent **M**anagement (see page 66)

HMAC **H**ash-based **M**essage **A**uthentication **C**ode (see page 68)

IT **I**nformation **T**echnology (see page 69)

MFA **M**ulti-**F**actor**A**uthentication (see page 61)

MitM **M**an **i**n **t**he **M**iddle (see page 61)

UK **U**nited **K**ingdom (see page 62)

GCHQ **G**overnment **C**ommunicatons **H**ead**q**uarters (see page 62)

CERT **C**omputer **E**mergency **R**esponse **T**eam (see page 62)

APTs **A**dvanced **P**ersistent **T**hreat**s** (see page 62)

DNI **D**irector of **N**ational **I**ntelligence (see page 62)

OSINT **O**pen **S**ource **Int**elligence (see page 62)

MO **M**odus **O**perandi (see page 62)

EDR **E**nd-Point **D**etection and **R**esponse (see page 63)

OS **O**perating **S**ystem (see page 74)

Chapter 1

Introduction

1.1 Introduction

In the past few years we've been witnessing an exponential growth in the number and intelligence of cyber attacks[1][2], most of them with the specific purpose of obtaining Personally Identifiable Information (PII) in an unauthorized way in order to cause damages, both monetary and reputational to either Companies, Governments or single individuals[3][4].

Despite the constant technological advances and the appearing of novel technologies almost every day, systems and users are still very vulnerable to these kinds of attacks. In this Internet of Things (IoT) era, where everything is connected and where data portability is considered an asset [5][6] to everyone, data leakage has become a serious issue. In fact, nowadays, not only businesses rely on data[7], but also social relations and personal interaction rely on it through social media and online platforms, all to an extent that human judgment may be overruled by data-driven decisions [8].

We all access and share online content like news posted on a digital newspaper or website, movies, short films or songs made available on online platforms like Netflix, Youtube or Spotify, and all of these through social media platforms like Facebook, Twitter and Instagram, or through instant messaging services like

Messenger, WhatsApp, Signal or Telegram, either from know people and friends, but also from unknown and possibly undesired individuals, therefore, being able to securely access our private data anywhere and share it only with authorized entities, while keeping it securely stored, is a subject of great matter.

Currently, individual users and companies depend and rely on public storage cloud providers to store their most sensitive data. The current security models used by these public cloud providers is, most of the times, not user-centric. This means that the users delegate their data and rely on the security offered by the cloud provider. However, as better as the security measures of the cloud provider data center might be, there is always the risk that a massive data breach on the cloud provider may disclose their customers data. There is also the risk that the cloud provider itself may become the attacker (for instance, motivated by political or competition reasons) and access on a non-authorized fashion the costumer's data. There are already some public cloud storage providers that encrypt the costumer data on their data centers, however the custody of encryption keys is, in some cases, unclear while on others they are on the cloud provider side, and this does not solve the problem presented before. There are also some public cloud storage providers that already offer the control of the encryption keys at the user side, however they rely on vertical solutions that do not allow to spread their encrypted content over different public cloud providers nor offer the appropriated transparency and accountability to the controlled multi-cloud encryption and decryption processes.

The purpose of this work is to design and present a system based on Blockchain and Distributed Ledger Technologies (BDLT), for the implementation of a Distributed Multi-Cloud Secure Storage System (for short, DMCS^3), as a way to mitigate the above described issue. The purpose of the proposed system is to allow the secure and distributed storage of data across multiple public cloud providers while preserving its confidentiality and integrity properties throughout its entire life cycle, while it enforces redundancy and reliability of the stored data.

1.2 Research Questions and Objectives

It is intended with this work, on a first step, to present a feasible and viable solution in terms of increased data security, accessibility and portability, later on followed by a comparison between this solution and actual solutions and systems. For this purpose, a comparative analysis will be made between the proposed solution and similar or identical solutions, targeting possible threats with different types of attack scenarios in order to assess how both this and other solutions are expected to behave in similar and comparable circumstances.

also, and in order to answer the question "Does using BDLT help to improve data confidentiality, integrity and availability in multi-cloud environments?", after the above demonstration, an assessment will be made regarding the implementation of BDLT on the previously proposed solution. It is also intended to evaluate it's security enhancements regarding other known solutions, in order to prove that a totally decentralized, peer-to-peer system, is an effective solution to store data in a distributed and multi-cloud system, without compromising its confidentiality, integrity and availability.

1.3 Methodology

Following the Design Science Research in Information Systems, this work sill start with an identified problem with improvement opportunities in the organisational context, and taking into account all the technology involved in this context, theoretical models and methods will be combined with empirical research methods in order to design a process that will later on be evaluated in a case study which will compare theoretical behaviours between similar systems [9].

A theoretical model which will address the identified relevant problems will be presented and all the research that provides clear and verifiable foundations and contributions will be fully addressed in order to demonstrate the model's utility, efficacy and quality [9].

1.4 Document Organisation and Structure

This works starts on the present chapter with the Introduction, and is followed by 4 more chapters.

In chapter 2 it will be presented the relevant related work for this Thesis, subdivided into 3 sections. The first section, regarding Cloud Storage and Distributed Systems will address issues like data security and data availability in Cloud environments and is followed by section 2.2 which addresses both Information Security and its core principles, and relevant Cryptography solutions for the work that will follow in chapters 3 and 4. Section 2.3 will target Blockchain Technology and its principles, while addressing relevant security and risk issues and presenting some applications in use today that are conceptually similar to the solution proposed in this work.

Chapter 3 will present the conceptual design of the proposed solution while addressing all its intended objectives and what is expected from it, while in chapter 4 the results from the previous chapter will be analyzed and discussed by means of comparison studies between similar solutions already available to the public, in order to access if the intended objectives and goals are met or not. Conclusions about this work and recommendations for future work will be set in the last chapter.

Chapter 2

Related Work

In the present chapter a review of the different relevant subjects of interest for this work will be presented, in order to better detail the different technologies, approaches and methodologies that shall be used in the next chapters. Furthermore, for each and one of the topics it will be shown its relevance for the proposed solution regarding digital data storage, security and availability.

Starting by addressing Cloud Storage and Distributed System's different delivery models, characteristics, components and key assets, Data availability considerations ans security issues in Cloud systems will follow, while presenting evidence of unintended outcomes regarding outages and security incidents and presenting a multi-cloud approach comparison.

In section 2.2, relevant work in the areas of information security and cryptography will be presented regarding system's confidentiality, integrity and availability, and several cryptography topics and methods will be also covered. The present chapter will end with Blockchain and Distributed Ledger Technology, it's core principles, types of blockchains, and blockchain applications that are considered relevant to this work or are worth to emphasise due to it's core characteristics.

2.1 Cloud Storage and Distributed Systems

As it is intended with this work to propose a solution for multi-cloud secure digital storage of information on public Cloud Systems, it is of the best interest to clearly identify Cloud Computing as a model that allows access to an online network of shared and configurable computing resources in a convenient and on-demand way, and where these resources can be rapidly provisioned or released with minimal effort and without a service provider interaction [10]. This model in fact, may be dissected in a few components given it's characteristics, delivery models and deployment models as shown in Figure 2.1.

Out of the 3 delivery models presented, some authors define Infrastructure as a Service (IaaS) , as a service on which the cloud provider supplies a set of virtualized infrastructural components such as virtual machines (VMs) and storage on which customers can build and run applications [11], while in a Platform as a Service (PaaS) , after building and running their applications on the platform, users are responsible for taking care of them while cloud providers assure the protection of those applications from other users." [11]. Software as a Service (SaaS) on the other hand, is defined as a software delivery paradigm in which the software is hosted off-premise and delivered via web [12].

Nevertheless, in public clouds, all of the three above share the commonality that the end-users' digital assets are taken from an intraorganizational to an interorganizational context, creating a number of issues, among which security aspects, which are considered to be one of the most critical aspects while considering cloud computing adoption [13].

Even though some authors state that there are only two types of Cloud Infrastructures, Public and Private [15], others state in their works that cloud deployment models include both public, private, community, and hybrid clouds [11] [10], with the National Institute of Standards and Technology (NIST) clearly specifying the differences between them and while key assests of Cloud Computing are identified as to be [16]:

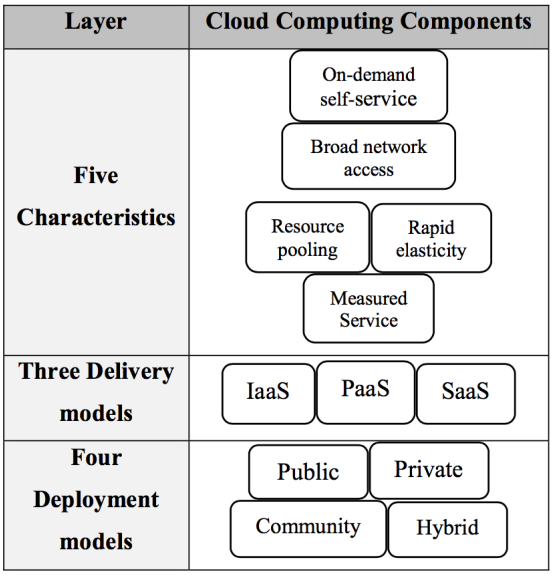

FIGURE 2.1: Cloud Environment Architecture [14]

1. Flexibility/Elasticity
2. Scalability of infrastructure
3. Broad network access
4. Location independence
5. Reliability
6. Economies of scale and cost effectiveness
7. Sustainability

Therefore, and according to each user's specific personal and/or business needs, a different Deployment Model might be applied for each and every one of the 3 Delivery Models, depending on the targeted characteristics.

According to Forbes (2015), by 2020 roughly 1,7MB of data will be generated every second by every human being [17], and knowing that a big part of this data comes from mobile devices people use everyday [18], with big business (and personal) impacts [19] [20] [21], it is of the best interest for the companies who sell storage space to provide an efficient and secure service. Additionally, security is

one of the biggest barriers in cloud technologies adoption by businesses [22] [23], and this problem needs to be addressed.

2.1.1 Data Availability in Cloud Systems

Data availability is one of the biggest concerns regarding cloud storage because system outages or failures can lead to major disorders with big potential losses if the data is not timely available. In fact, some of the current Cloud Storage Providers (CSPs) do not offer any guaranties regarding long-term availability or security meaning that our data could be permanently lost, or violated in terms of security and therefore, to face some of the issues regarding long-term data availability in the clouds. Celesti et al. [24] propose in their work a system that allows its users to rely on different CSPs while enforcing long-term availability, obfuscation and encryption. Based on these assumptions, and combining them with the fact that if a certain CSP is temporarily unavailable or not permanently available, end-users will still continue to access their data in a secure way, while only them will have the full control of the its security and without disclosing any sensitive information to any CSP.

Another important feature of Cloud Storage is its efficiency, both for CSPs and end-users, and Papaioannou, Bonvin and Aberer [25] present us in their work the Adaptive Scheme for Multi-cloud Storage Efficiency - SCALIA - a cloud storage mediation solution that, based on access patterns, systematically adapts the data positioning, being subject to optimization goals such as storage costs , targeting the re-positioning of selected objects if it significantly lowers the storage costs. The authors prove the cost-effectiveness of this solution by extensive simulation experiments, making it a alternative solution for single-cloud systems, while not compromising the assumptions stated above regarding [24] work, making it an interesting approach to multi-cloud storage when combining both of these approaches.

The Figure 2.2 below shows us a few examples provided in Rimal, Choi and Lumb's work (2009) regarding outages in Cloud Services, and as stated by Alvi, Qureshi and Karim [26], "*for any business to grow successfully the information availability plays a vital role leading to continuous uptime and minimum downtime, which is the mission of every successful business organization*".

Service and Outage	Duration	Date
Microsoft Azure: Malfunction in Windows Azure	22 Hours	March 13-14, 2008
Gmail and Google Apps Engine	2,5 Hours	Feb 24, 2009
Google search outage: programming error	40min	Jan 31, 2009
GMAIL: site unavailable due to outage in contacts system	1,5 Hours	Aug 11, 2008
Google AppEngine partial outage: programming error	5 Hours	June 17, 2008
S3 outage: authentication service overload leading to unavailablebility	2 Hours	Feb 15, 2008
S3 outage: single bit error leading to gossip protocol blowup	6-8 Hours	July 20, 2008
FlexiScale: core network failure	18 Hours	Oct 31, 2008

FIGURE 2.2: Outages in Different Cloud Services [27]

According to Check Point's 2018 Security report [28], regarding the year 2017, some of the most common cyber attacks, were Distributed Denial of Service (DDoS) attacks, Ransomware attacks, like WannaCry, Petya and NotPetya that led Maersk, FedEx and WPP, for example, to a worldwide disruption, invasive mobile malware and Botnet army recruitment. Also, malware exploiting zero-day vulnerabilities and phishing and spear-phishing attacks led to major data breaches like the ones that targeted, for example, XBox, Playstation, Equifax, and Uber, being this last one of particular relevance due to its nature; user accounts details were stolen from a 3rd party CSP, in this case, an Amazon Web Services (AWS)

account was hijacked, being Uber forced to pay 100.000 US Dollars to cover up the breach.

Also, Accenture's 2017 Cost of cyber crime study refutes Check Point's report when stating that Ransomware attacks alone take a 27% slice of all cyber attacks. When combined with the remaining 73%, and only in the last year, the global average cost of cyber crime saw an increase of more than 24% [29], so it is then clear from this report that out of the most common attacks, the majority of them is aimed to cause either data or system's downtime and unavailability, leading to tremendous reputational and monetary losses. In fact, taking a closer look at Figure 2.3 from [26] work from 7 years ago, evidence can already be seen that, even if not intended by an attacker, security incidents led to unprecedented monetary losses.

Name	Impact
Morris Worm	Stopped 10% of computers that were connected to the Internet.
Melissa Worm	100,000 computers infected in one week causing $1.5 billion loss
Explorer Virus	$1.1 billion loss
Love Bug Virus	$8.75 billion Loss
Sircam Virus	2.3 billion computer infected causing $1.25 billion loss
Code Red Worm	359,000 computer infected in <14 hours causing $2. 75 billion loss
Nimda Worm	160,000 computer infected causing $1.5 billion loss
Klez	$750 million loss
BugBear	$500 million loss
Badtrands	90% of vulnerable hosts infected in just 10 minutes, $400 million loss
Sapphire/Slammer worm	75,000 hosts infected causing $1.5 billion
Blaster	$750 million
Nachi	$500 million
SoBigF	Fastest spreading mass-mailer worm causing $2.5 billion loss
MyDoom Worm	More then $4.0 billion loss due to 100,000 instances of the worm/hour
Witty Worm	First wide-spread worm to carry destructive payload

FIGURE 2.3: Security Incidents and its Impacts [26]

2.1.2 Data Security in Cloud Systems

As not only data availability is relevant in Information Security (IS), also its confidentiality and integrity are subjects of great relevance and therefore concern.

Based on these principles, Bowers, Juels and Oprea [30] propose a "*distributed cryptographic system that allows a set of servers to prove to a client that a stored file is intact and retrievable*" - the High Availability Integrity Layer (HAIL).

This HAIL system, as stated by the authors, "*cryptographically verifies and reactively reallocates file shares*" and it is robust enough against an adversary that may progressively corrupt the full set of servers [30].

Taking into account Bohli et al. work [13], where they state they have strong beliefs that data splitting and homomorphic encryption approaches seem to be "*the most viable alternative, both from the technical and economical point of view*", while defending that "*state-of-the-art encryption of data with adequate key management is one of the most effective means to safeguard privacy and confidentiality when outsourcing data to a cloud service provider*", and also other solutions [31] regarding cryptographic data splitting, a consensus can almost be found within authors that a very important feature of Data Security in Cloud Systems is it's partition and further encryption of its fragments.

In their work, Bohli et al. [13] present an assessment and comparison of multi-cloud approaches as shown in Figure 2.4, clearly identifying improvements and/or aggravations in each approach, regarding every aspect in study.

	Security			Feasibility			Regulation
	Integrity	*Confidentiality*	*Availability*	*Applicability*	*Business-Readiness*	*Ease of Use*	*Compliance*
Replication of Application							
Dual Execution	+	--	+	+	+	0	-
n Clouds Approach	++	--	++	+	+	-	--
Processor and Verifier	+	-	0	-	-	-	-
Partition of Application Systems into Tiers	+	-	0	++	++	++	0
Partition of Application Logic into Fragments							
Obfuscating Splitting	0	+	-	0	0	-	++
Trusted/Public Domain Splitting	++	+	0	-	-	--	+
Homomorphic Encryption	++	++	0	--	--	--	++
Secure Multi-Party Computation	++	++	-	--	--	--	++
Partition of Application Data into Fragments							
Cryptographic Data Splitting	++	++	-	-	0	-	++
Database Splitting	0	+	0	-	+	-	+

(++) strong improvement; (+) little improvement; (0) no change; (-) little aggravation; (--) strong aggravation

FIGURE 2.4: Comparison of multi-cloud approaches. [13]

On the other hand, other authors [32] propose the multi-clouds Database Model (MCDB), a model that uses multi-cloud Service Providers in order to target issues related to data security and privacy aspects in cloud computing, such as data integrity, data intrusion and service availability, widening the use capabilities of this system when combined whit the ones stated above.

Regarding DepSky project [33], the authors clearly defend this theory, while presenting a solution that is able to improve data confidentiality, integrity and availability through multi-cloud data replication and encryption. Their systems relies on an asynchronous distributed system which is composed by clients and multiple CSPs in order to replicate client's data through them, while hiding most of the distributed storage system's complexity.

Taking into account the works presented above, it can be assumed that as of today, data splitting through several CSPs with each of the fragments being encrypted in order to assure data security seem to be one of the most consensual approaches in order to guarantee a certain level of data security, while minimising its downtime and unavailability within the servers and CSPs due to data replication, even though it may increase the associated costs to such solutions. Overall, it is solemnly of the end-user's best interest to clearly identify what's most important to him.

2.2 Information Security and Cryptography

With the constant increase of data generated worldwide on a daily basis, with most of it being critical for both individuals, businesses and governments, information security breaches are a subject of big concern nowadays. Three very well known cases are the United States of America (USA) National Security Agency (NSA) breach [34], exposing Government Secrets to the public, causing great damage to the Government's reputation, the iCloud Hack [35], exposing thousands of intimate photographs and videos from worldwide personalities, damaging their reputation as well as the companies, and the Yahoo data breach in 2013 that affected 3 billion

accounts by exposing their names, e-mails, passwords, Tumblr, Fantasy and Flickr accounts [36].

Solms and Solms [37] present a very interesting approach on 10 topics they have found of most relevance for when companies are trying to implement an IS plan, based on their personal experience, clearly stating in one of the points that IS Governance is a multi-dimensional discipline while emphasizing on another point that an IS Plan must be based on identified risks. Even though a company has to deal with different kinds of IS breaches when compared to a singular citizen, risks however, remain the same, as for both cases the IS is dissected into four major components: assets, threats, vulnerabilities and impacts [38]. Nevertheless, there are several aspects on which they share a common interest regarding their information, like their privacy (users) and their most valuable data's privacy, moreover if this data is stored in a Trusted Third Party (TTP) like a CSP.

This brings us back to an old topic in several areas, like IS: Trust, which has a very interesting approach with the development of the Trusted Computer System Evaluation Criteria (TCSEC) in the end of the 1970's decade, where trust was used to convince observers that a system was correct and secure [16]. In fact, in the security solution proposed by these authors, trust and security threats identification are both the focus of the solution proposed for Cloud Security, where clients are leveraged from the security burden when trusting a TTP. Trust is nothing but when an entity believes another entity will behave precisely as expected and/or required, and when talking about security, it is not easy to trust anything we don't see or believe. For a CSP to be trustworthy, security issues need to be targeted first and before an application or system is developed or deployed. Subashini and Kavitha (2010) present us in their work 14 security issues that shouldn't be taken lightly when considering either to develop or to deploy a SaaS application, being:

1. Data security
2. Network security
3. Data locality

4. Data integrity
5. Data segregation
6. Data access
7. Authentication and authorization
8. Data confidentiality
9. Web application security
10. Data breaches
11. Virtualization vulnerability
12. Availability
13. Backup
14. Identity management and sign-on

while Zissis and Lekas [16] provide us a few guided steps on how to identify unique challenges and threats regarding confidentiality, integrity and availability.

Figure 2.5, present in their work, give us a brief overview of the security complexity existing in a Cloud Environment, regarding its different layers.

2.2.1 Information Security Principles

The fundamental question in IS is determining what needs to be protected, why it needs to be protected, what or whom it needs to be protected from, and how to protect it for as long as it exists [38]. There are several approaches to address this question like vulnerability assessments, IS Audit, IS Risk Evaluation and Managed Service Providers, and none of them will solve every problem in a single try, mostly because some of these approaches depend on others. For example, an IS Audit based on ISO/IEC 27001 requires, among others, vulnerability assessments and

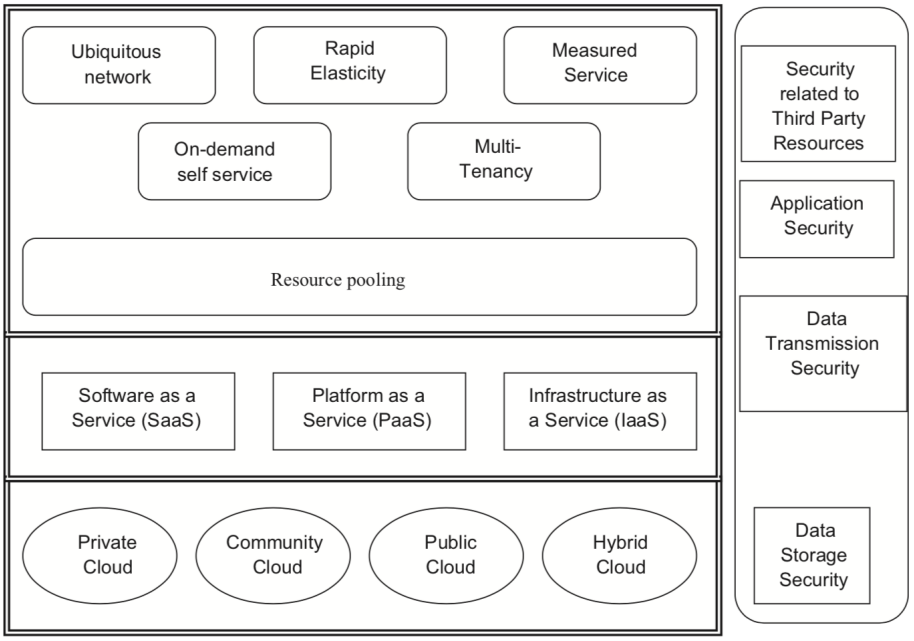

FIGURE 2.5: Security Complexity in Cloud Environments [39]

risk evaluation in order to be successful. While the OCTAVE method, which is based on a 3-phase approach to address technological and organisational issues with a defined scope in asset-based threat profiling, security strategy and plan development and infrastructure vulnerabilities identification [38], might be suited for companies and enterprises, for individuals it's too excessive and it's not applied, nevertheless, there are 3 key characteristics of IS that both share in common and on which multiple authors find a consensus: **Confidentiality**, **Integrity** and **Availability** (CIA).

The same authors describe, in their work [38], **Confidentiality** as "*the requirement of keeping proprietary, sensitive, or personal information private and inaccessible to anyone who is not authorised to see it*", **Integrity** as "*the authenticity, accuracy, and completeness of an asset*" and **Availability** as "*the extent to which, or frequency with which, an asset must be present or ready for use*".

In order to establish a correct approach to mitigate the problems associated with these three principles, the identification of the risks associated to them should be the first approach [37]. For this purpose we should know that our assets can be compromised by the existence of vulnerabilities, that can be, either intentionally or

accidentally, exploited, resulting in the destruction, damage or illegally obtaining the asset - **Threats** - [40]. Therefore it is important to understand which are the vulnerabilities affecting our assets.

After examining other works and after interviewing three Chief Information Security Officers (CISO) , Whitman (2003) presented us a very interesting work regarding threats identification. From his work 12 threat categories were identified and ranked as follows:

1. Deliberate Software Attacks
2. Technical Software Failures or Errors
3. Act of Human Error or Failure
4. Deliberate Acts of Espionage or Trespass
5. Deliberate Acts of Sabotage or Vandalism
6. Technical Hardware Failures or Errors
7. Deliberate Acts of Theft
8. Forces of Nature
9. Compromises to Intellectual Property
10. Quality of Service Deviations from Service Providers
11. Technological Obsolescence
12. Deliberate Acts of Information Extortion [41]

As for vulnerabilities identification, it is recommended the use of vulnerability sources to identify system vulnerabilities, as well as the performance of system security testing, and the development of a security requirements checklist [42]. From these, the authors defend that both vulnerabilities and threats can be identified in order to reduce or eliminate risk during the risk mitigation process [42].

Regarding data availability, which might be critical to some businesses [43], we can identify:

1. Physical and Logical Security
2. Information security policy
3. Operational control processes
4. Pro-active hardware management
5. Hot spares' inventory
6. Tested and certified configuration
7. Load balancing and configuration management
8. Continuous system monitoring and inspection
9. IT auditing and system effectiveness evaluation
10. Hardware redundancy
11. Data backup
12. Business continuity [26]

These are the key determinants of information availability identified during the course of their research, while subdividing them into three components - Reliability, Accessibility and Timeliness - and proposing a model of information availability that organizations may adopt, according to their goals, aims, objectives and business needs [26].

Regarding information integrity, Boritz created a framework [44] based on a literature review on data quality and information integrity, while gathering at the same time experienced IS practitioners' views on issues like its core attributes and enablers, its relative importance and relationships between information integrity attributes and enablers through a questionnaire. According to the author, "*integrity is the representational faithfulness of the information to the condition or*

subject matter being represented by the information" [44]. Nevertheless, in the definition provided by ISACA (2000), information integrity is defined by three attributes combined: completeness, accuracy and validity.

Figure 2.6 below, depicted in Boritz's work [44], shows how information integrity is enhanced by processing integrity, while emphasizing its dependence on a system's availability and security.

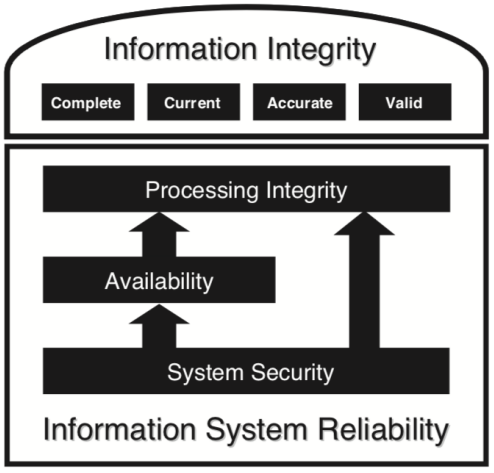

FIGURE 2.6: Relationship between information integrity, processing integrity and system reliability [44]

As for Confidentiality, the OCTAVE referred previously, regarding its meaning, clearly states what this attribute represents. Besides companies, to whom this approach is more intended to, the importance of information confidentiality in individuals might also be critical, specially when referring to PII. McCallister, Grance and Scarfone (2010) categorize in their work the different types of PII while defining it as "*any information about an individual maintained by an agency, including (1) any information that can be used to distinguish or trace an individual's identity, such as name, social security number, date and place of birth, mother's maiden name, or biometric records; and (2) any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information*" [45]. Through various examples throughout their work, the authors define the types of impact levels based on the harm caused by a breach of confidentiality as well as the factors for determining them, while offering safeguard

approaches, security controls and incident responses companies or services should take when there is a PII Confidentiality breach.

In fact, and as stated above, any confidential data should be kept private and inaccessible to anyone who is not authorized to see it, either in rest or stored. In order to achieve such objective several methods may be applied. One of the most common, in use for more than 4000 years, is Cryptography.

2.2.2 Cryptography

From the Egyptians to the modern days, cryptography has been playing a huge role in Diplomacy, Politics and Military sectors, providing means for pieces of information to be kept secret from undesired eyes. Besides confidentiality and integrity assurance, cryptography can also help attain Authentication and non-Repudiation principles, being its fundamental goal to "*adequately address these four areas*" [46].

In their work [46], the authors present several different cryptographic tools (which they describe as "primitives") in order to ensure information security. These primitives are depicted below (Figure 2.7) and should be evaluated according to the following criteria:

1. Security level
2. Functionality
3. Operation methods
4. Performance and
5. Implementation easiness [46].

Nevertheless, the authors defend that the importance of each and every one of these criteria should be taken into account, depending on the application and

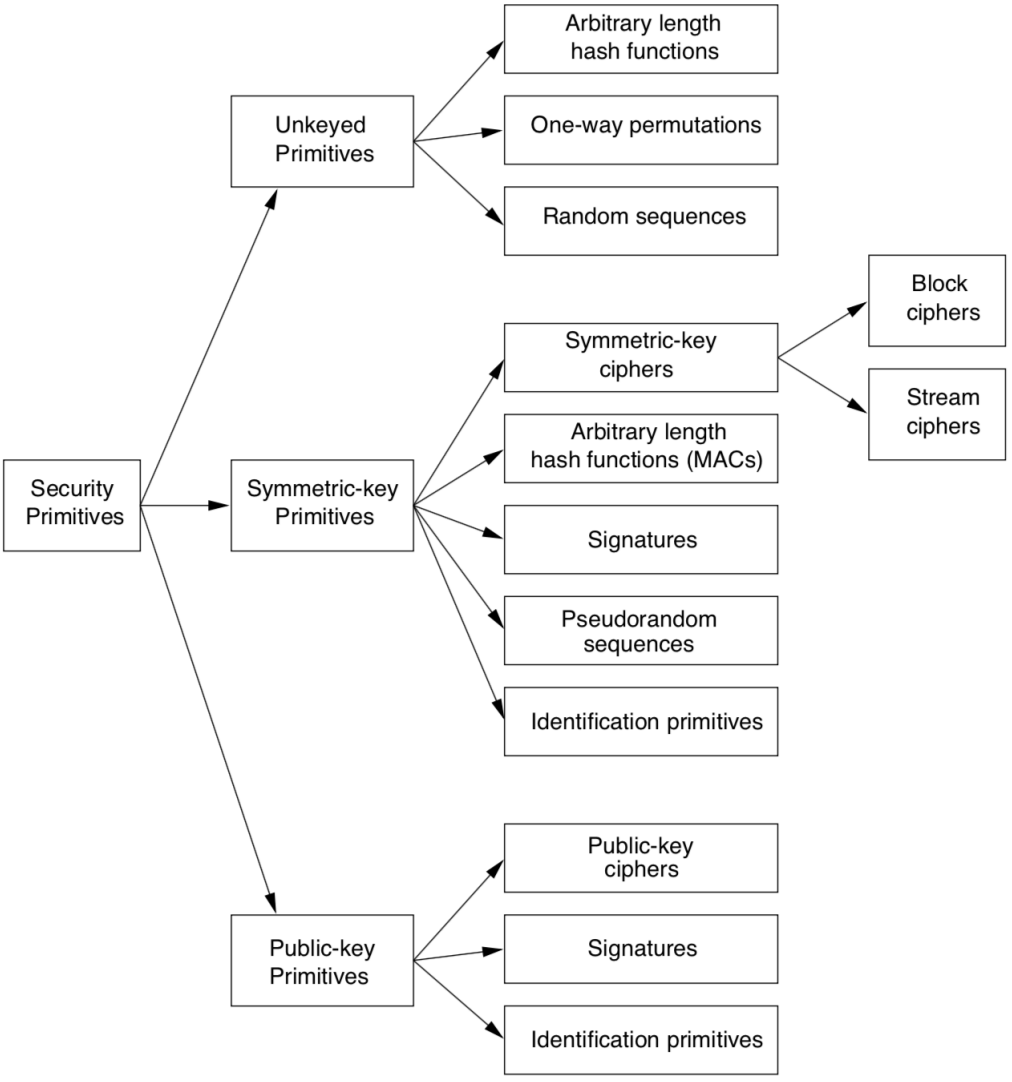

FIGURE 2.7: Cryptographic Primitives [46]

resources available. For instance, if our computing environment has an outstanding performance, a weaker performance might be traded off for a higher level of security.

For all these primitives, a mathematical approach is provided in their work, as well as examples and explanations of how those primitives work. Also, public key encryption and hash functions are fairly approached in their work.

Public Key Cryptography, also called Asymmetric Cryptography, is a cryptographic system that uses a pair of keys ("public" and "private") in order to assure data authentication and encryption. Lets assume user A and user B: User A wants

to send a message to user B but there is no secure channel to do so. Knowing that each user as a set of keys and that only the private key can decrypt the message encrypted by the public key (and vice-versa), as long as user A encrypts the message with user B's public key (which can also be transmitted through an unsecured channel), only user B's private key can decrypt it. Of course a private key has to remain private, and as long as it remains like this, confidentiality can be assured. Nonetheless, we can not verify the authenticity of the message unless user A encrypts it first with his private key. This means that when user B receives the message, only him can decrypt the first "layer" (with his private key) assuring the integrity and confidentiality of the message, but also the authenticity of the message can be verified if later decrypted with user A's public key, assuring that only him could write it and send it, because only him had his own private key.

Some of the advantages of Public-Key Cryptography, as stated by the authors, are:

1. Only one key out of the two must be kept secret
2. The Key administration on a network requires only one TTP
3. In a large network, the number of keys necessary may be considerably smaller than in the symmetric-key scenario [46].

As for disadvantages of Asymmetric Cryptography, the authors defend that Key sizes are typically much larger than those required for symmetric-key encryption. Also, according to the authors, no public-key scheme has been proven to be secure, being its assumed security been based on presumed difficulty of a small set of number-theoretic problems [46].

Figure 2.8 below provides us an insight on how public-key cryptography works.

Another fundamental primitive in modern cryptography is the cryptographic Hash Function, a function capable of mapping binary strings of arbitrary length to binary strings of some fixed length, called hash-values [46], i.e, they return a specific output value, with the exact same number of characters despite the

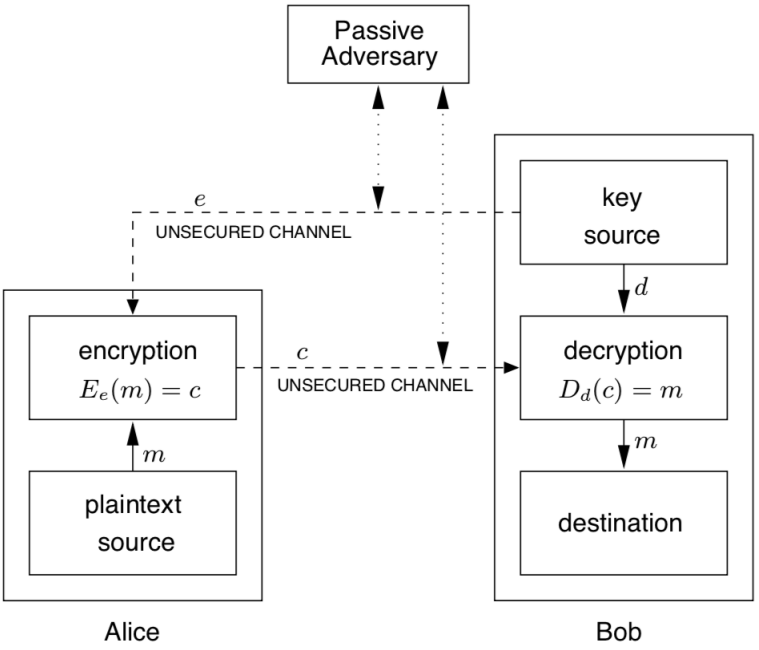

FIGURE 2.8: Encryption using public-key techniques [46]

input value's size, meaning that as long as there are no different characters in an initial string, regarding the strings size, the output will always be the exact same. But if, on the other hand, one single character is changed in, for example, a 500 character sentence, the output string will be totally different from the first one in terms of characters, but never in size. Also, if we left only 1 character out of those initial 500, the output would behave the exact same way as described before. There are no ways to reverse a hash function, and therefore, the most common way to enforce a bypass is by a hash collision, which is only possible in a few hash functions, like SHA1 for example, but not all. Hash functions are most commonly used for hashing user's passwords and ID's in order to maintain their confidentiality and integrity, being lodged in databases, so that when a user tries to login to a service the hashed values put as inputs are the ones that are being compared with the ones stored for user access authentication.

Digital signatures and data integrity verification are the most common cryptographic use for Hash Functions, being these publicly known and involving no

secret keys (for example Modification Detection Codes if used to detect if a message input has been altered), however, they can also involve secret keys and provide data authentication as well as data integrity (for example Message Authentication Codes) [46].

Its main properties are:

1. **Preimage Resistance**
2. **2nd Preimage Resistance**
3. **Collision Resistance**

Preimage resistance refers to the fact that the cryptographic hash function is non-reversible, i.e, it is impossible to turn a hash value into the original string, and 2nd preimage resistance on the other hand, is the hability of the function to always return a unique output value despite its input. Although this is a similar concept to collision resistance, the main difference between the 2 is that in 2nd preimage resistance we try to find a second input that generates the same output as the first, given input, while in collision resistance we try to find 2 different inputs that generate the same output.

Figure 2.9 depicts the classification of Cryptographic Hash Functions and applications.

Besides these two Cryptographic approaches mentioned above, several others with much relevance like Symmetric-Key Encryption, Digital Signatures, Block Ciphers and Stream Ciphers are approached by the authors, as well as the algorithms efficiency for almost every approach is studied, while emphasising key protocols and key management techniques.

Nonetheless, some authors present us in their work the current state of Multivariate Public-Key Cryptography - asymmetric cryptographic primitives based on multivariate polynomials over a finite field - while providing insights on how they are constructed and how they can achieve NP-Completeness, being able to resist quantum computer attacks. The four examples provided are:

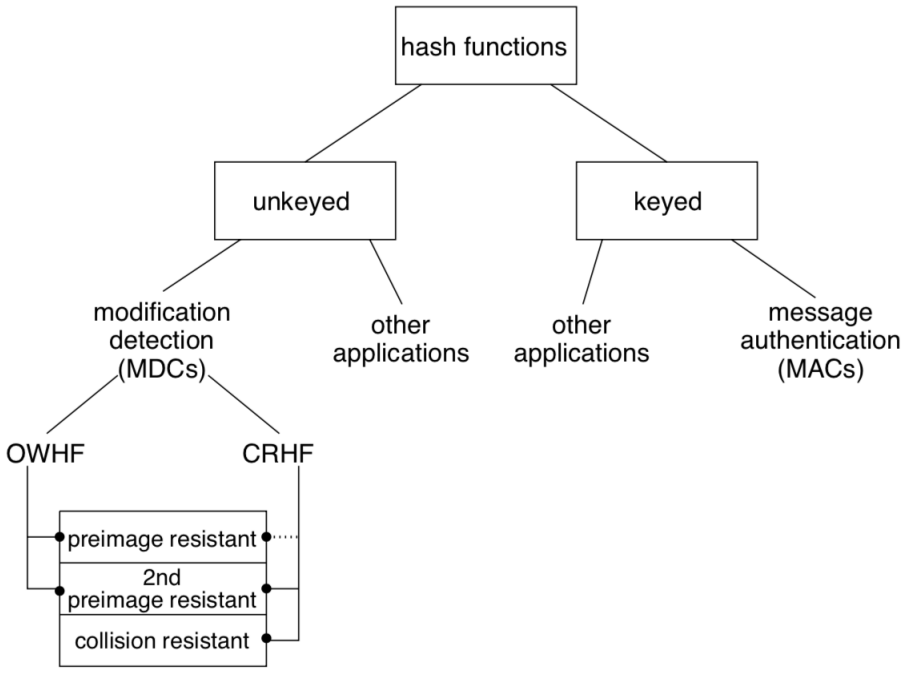

FIGURE 2.9: Simplified classification of cryptographic hash functions and applications [46]

1. hash-based signature schemes of the Diffie-Lamport-Merkle type
2. lattice-based public-key cryptosystems
3. code-based public-key cryptosystems (in particular, the McEliece encryption scheme)
4. Multivariate Public-Key Cryptosystems (MPKCs) [47]

Also, and despite not being very mentioned in the above mentioned work [47], elliptic curve cryptography (ECC), which is also a public-key cryptography, is also worth to mention due to its mathematical operations that are instead defined over the $Y^2 = X^3 + AX + B$ curve where $4A^3 + 27B^2 \neq 0$ and where the public-key is a point within the curve generated by multiplying a random number, i.e, the private key, by the curve parameters A and B, which return a different elliptic curve every time. Due to its characteristics and mathematical operations, it is emphasised in [48] that a 160-bit key in ECC is as secure as an RSA 1024-bit key.

On the other hand, and regarding cryptographic security to protect data storage systems, some authors make important comparisons [49] between Public-Key and Symmetric-Key approaches, while clearly defining the incurring challenges for properly protecting storage systems. In [49], the challenges identified are:

1. Rapid increase of sensitive data
2. Storage Systems and devices are networked and distributed
3. There are several attack points during data's life-time
4. Government regulations require long term data retention and protection
5. Compatibility
6. 24/7 continuous data protection and
7. Storage system should have acceptable usability, manageability and performance [49]

2.3 Blockchain and Distributed Ledgers Technology

Another approach to methods to ensure Data Protection, besides the ones mentioned above, is to decentralise privacy in order to protect personal data [50], like recurring to Blockchains and Distributed Ledgers.

Distributed ledgers can be seen as a decentralised database that relies on every node (or computer) in the network to perform its operations, like record, share and synchronise transactions, and ultimately, to store its data.

A Blockchain is just a type of distributed ledger, i.e, an "*open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way*" [51]. First idealized and proposed by Satoshi Nakamoto for its purely peer-to-peer version of electronic cash - "Bitcoin" [52], Blockchain

technology is based on several principles like being a totally distributed system and synchronized across networks. There is no need for a TTP because all the network works on a consensus (i.e.: in order for a transaction to be considered legitimate, the whole network has to agree on it, and not just a single node). The data is immutable; once all nodes in the network agree that a transaction is legitimate and it happens, it can never be changed again. Also, because of this property, it is possible to know the provenance of every bit of information; where it came from, and where it is at any time.

For every transaction on the Blockchain, as proposed by Satoshi Nakamoto, several requirements need to be met, and the use of both Asymmetric Cryptography and Hash-Function Cryptography mentioned in the previous section are clearly depicted in figure 2.10 below.

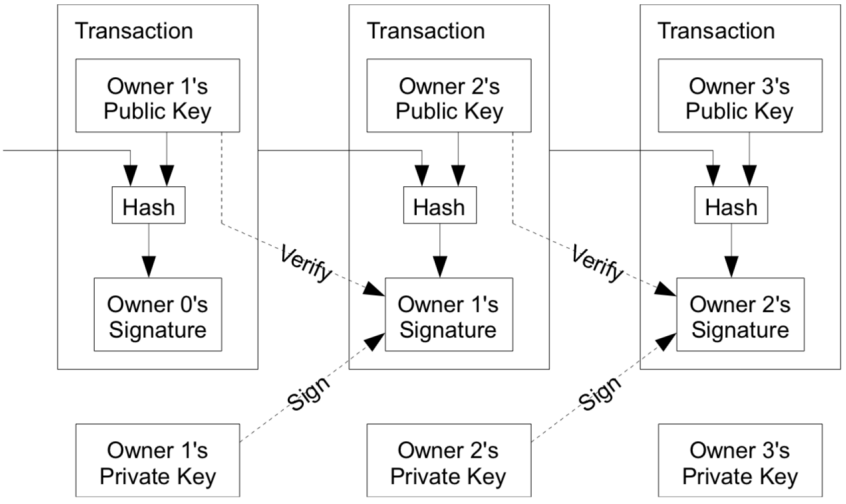

FIGURE 2.10: Bitcoin Transaction [52]

The schematics below (Figure 2.10) ensures that every piece of information is not lost, since for every transaction recorded an hash with the information from the last transaction combined with the actual owner's public key is created, and then signed by the previous owner's private key (which is also verified by the previous transaction's public key). The problem with this solution is, if we are talking about money transactions (like Bitcoin), it cannot be checked if an owner "double-spends" the money at the same time unless there is a TTP (like a bank, for

example) or a mint, even though these would take apart the intended decentralized solution, and therefore, a "Timestamp Server" is proposed.

This Timestamp Server solution proves that some data must have existed at the time so it can get into the hash, and every timestamp includes in its hash the previous timestamp, with each additional timestamp reinforcing the ones before it [52].

To implement a distributed timestamp server solution on a Peer-to-Peer (P2P) system, a proof-of-work system like the one in figure 2.11 below is proposed by Nakamoto [52]. This proof-of-work involves scanning for a value that when hashed, the hash begins with a number of zero bits. For the timestamp network, the proof-of-work adds a nonce in the block until a value is found that gives the block's hash the required zero bits [52].

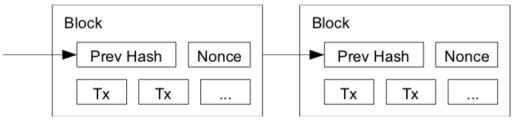

FIGURE 2.11: Blockchain Proof of Work [52]

After a block is completed, it is added to the previous block, making the "chain" bigger with time. This could bring the problem of building alternative chains in order to subvert the previous actions, and therefore, being a P2P system, a consensus between nodes needs to be achieved in order to the transactions be considered legitimate and to the next blocks to be added to the previous ones in the chain. This is achieved by assuring the proof-of-work system is based on a "One-CPU-One-Vote" system, and by assuring that the different nodes in the network always consider the longest blockchain to be the correct one.

As proposed by Nakamoto in his work, the network has to run in 6 steps:

1. All the new transactions have to be transmitted to all nodes in the network
2. Each node collects those transactions into a block

3. Each node finds a proof-of-work for its block
4. When the node finds it, broadcasts the block to all nodes
5. Nodes accept the block only if all transactions in it are valid and not already spent
6. Nodes accept the new block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash

Of course, Security and Performance issues may arise with every system, and blockchains are no different. Some of the most common ones are the so-called "double-spending", on which a user tries to deceive the blockchain by double-spending its currency (Bitcoin, Ethereum [53], Litecoin [54], etc), "selfish mining", on which attackers intended to increase their relative mining share in the blockchain, ultimately trying to gaining control of it, Border Gateway Protocol, Eclipse, Liveness and Balance Attacks. All of these attacks and performance issues of different blockchain protocols are approached in [55] as well as in [56] and [57]. Also [58] present us their work regarding this subject, with more emphasis on performance characteristics, compliance and trust issues , while presenting us in their work several IS characteristics of different Blockchains, shown in figure 2.12 below.

Due to its nature, many of the key applications for blockchains target security issues in decentralized platforms [59], and the author evaluates blockchain's roles in protecting privacy while strengthening cybersecurity both in IoT platforms, healthcare industry, data storage and others. Also targeting security in blockchains,other authors made a survey to address this issue while presenting some risks to blockchains, attack cases like the ones described above and security enhancements like "Smartpool" and "Quantitative Framework" [57].

It is also important to distinguish the types of Blockchains available, since they all don't work exactly the same way. For instance, the blockchain type targeted above is the public one, since the decentralized ledgers are accessible to every internet user and its nature derives from the "*free and unconditional participation*

Many Different Types of Blockchains						
Principle	**Bitcoin**	**Ethereum**	**Stellar**	**IPFS**	**Blockstack**	**Hashgraph**
Confidentiality	None	None	None	Hash-based content addresses	None	None
Information availability	Block mirroring	Block mirroring	Ledger mirroring	Graph and file mirroring	Block mirroring/ DHT mirroring	Hashgraph/ mirroring; Optional Event History
Integrity	Multiple block verifications	Multiple block verifications	Latest block verification	Hash-based content addressing	Multiple block verifications	Consensus with probability one
Non repudiation	Digital signatures	Digital signatures	Digital signatures	Digital signatures	Digital signatures	Digital signatures
Provenance	Transaction inputs/outputs	Ethereum state machine and transition functions	Digitaly signed ledger transition instructions	Digital signatures and versioning	Transaction inputs & outputs and virtual chain references	Hashgraph/ mirroring; Optional Event History
Pseudonymity	Public keys	Public keys and contract addresses	Public keys	Public keys	Public keys, but public information encouraged	Not supported; could be layered
Selective disclosure	None	None	None	None	Selective access to encrypted storage	Not supported; could be layered

FIGURE 2.12: IS Principles on Blockchains [58]

of everyone in the process of determining what blocks are added to the chain" [60]. On the other hand, in a fully private ledger write permissions are controlled by a central entity in terms of decision-making and read permissions can be either public or restricted, making the main difference between both public and private blockchains the extent on which how much they are decentralized and how much they ensure anonymity [60].

Even though they are pointed as a revolution [61] [62] and a golden nugget in technology, Blockchains are not the solution for every problem, but nevertheless both companies and individuals are getting more and more familiarized with this technology and knowing its potential, so it is expected that those with more financial power invest in this new technology in order to face new technological challenges, both in the financial sector, health care records management, supply chains, property titles, online identities and so on [63]. In order to let us better

understand if we really need a Blockchain, a flowchart like the one depicted below in figure 2.13 was introduced.

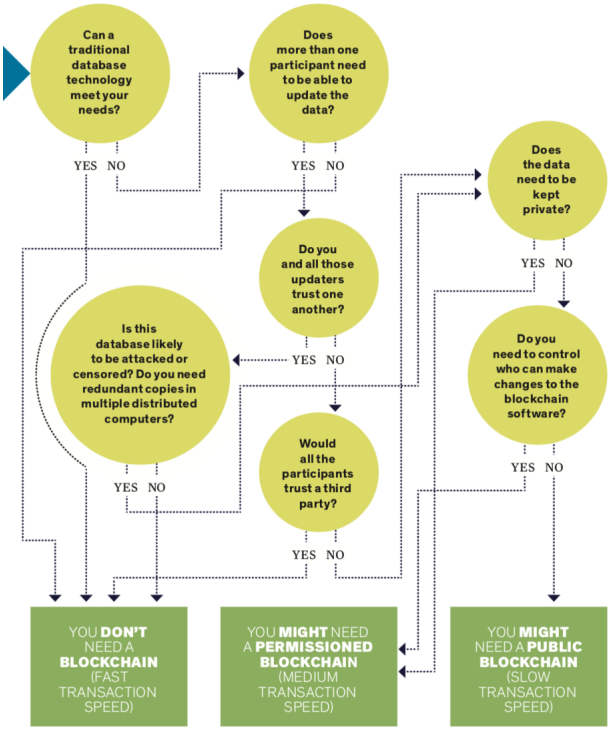

FIGURE 2.13: Blockchain need flowchart [63]

2.3.1 Blockchain Applications

The first applications that come to our thoughts when Blockchain is mentioned are, almost inevitably, Cryptocurrencies and Smart Contracts. Cryptocurrencies are digital currencies who use cryptography in order to enforce security. These can not be counterfeit, and there is always a record of every transaction on the Blockchain, making it a groundbreaking revolution in the financial markets, with already some countries on the line to create their own virtual coin [64]. It is estimated that this market alone had surpassed the 500 Billion Dollar mark as of December 2017 [65].

Smart Contracts are also another groundbreaking application based on Blockchain Technology, while they eliminate any third party in a negotiation or exchange of information/asset, while executing a contract with the predefined terms agreed

between the two parties involved. All of these transactions are also traceable, transparent, and irreversible, opening the possibilities for trading financial assets, making insurance policies or even to buy goods.

E-Voting systems are also one of the applications of this technology, while providing full transparency to the electoral act, since every single vote is recorded on the Blockchain, if built "*upon the immutability, transparency and consensus inherent to the blockchain technology*" [60].

Regarding Data Storage and PII Security, numerous applications for this technology can be found. MIT Researchers proposed the "ChainAnchor" as a new way to establish a privacy preserving and trusted identity, by adding an identity and privacy-preserving layer above the Blockchain [66]. With "Enigma" for instance, "*data is split between different nodes, and different nodes cooperate to compute functions together without leaking information to the other nodes*" [66], meaning that every single party only has a small part of the information and not all of it. Also STORJ, who claims to be the most secure and private cloud platform, works on a Peer-to-Peer basis with End-to-End Encryption and High-Availability [67], while offering very competitive prices for cloud storage services. In other words, data is stored in the nodes within the network, i.e., other people's devices, being rewarded with a compensation for having their system available to store data and share computing processing power in order to solve Ethereum's blockchain cryptographic equations (mining power).

In figure 2.14 below is shown the Metadisk Model of Data Storage: The non-technical User Interface (UI) and the development platform for the STORJ Network.

Besides these data storage solutions, there's also Boxcryptor and Tresorit, who provide end-to-end encryption file sharing and synchronisation, being the main difference between them the CSP. If the first, Boxcryptor, relies on 3rd party CSPs in order to store data while enforcing its encryption within the application and before the upload to the CSP [69], Tresorit on the other hand, has a proprietary

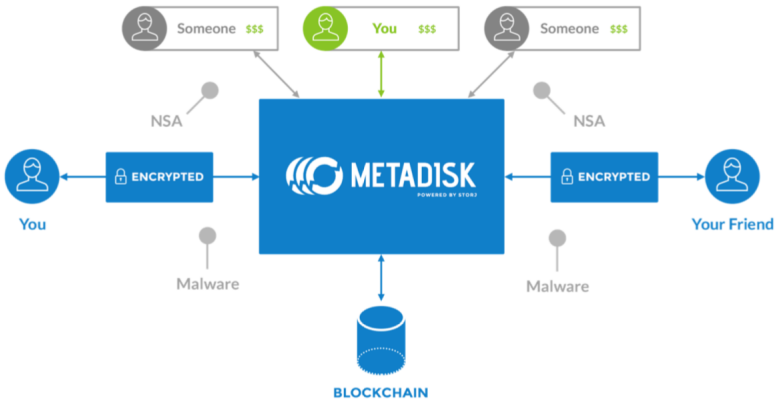

FIGURE 2.14: Metadisk Model of Data Storage [68]

cloud service and offers the full service by their own without relying on other CSPs [70].

Cloud Storage solutions available nowadays are countless, mainly differing between them on the chosen CSP (proprietary or 3rd party), the communication channels and data encryption types, and the storage distribution type, i.e, if it's a distributed storage or not, relying on single or multiple CSPs

It is unquestionable that the solutions and opportunities BDLT brings us are almost uncountable, and can be applied to every aspect of our modern, IoT and data driven society, therefore, and taking into account the amount of connected devices a single user has at its own disposal, who generate, according to Forbes, roughly 2,5 quintillion bytes of data every day through, for example, Social Media platforms, digital photographs and IoT devices [71], it should be considered as of utmost importance to keep user's data private and available only to the desired audience, while being capable of guaranteeing its authenticity and non-repudiation from third parties, all of which are achievable recurring to BDLT. Taking this into account, and combining BDLT with secure and distributed cloud storage that relies on multiple encryption techniques in order to achieve data confidentiality, integrity and availability should open the doors to different approaches and discussions regarding this topic, which will be addressed during the course of this work.

Chapter 3

Proposed Solution

The information security solution suggested in this work intents to create the means for a distributed multi-cloud security storage solution intended to provide the confidentiality, integrity and availability of data assets in the cloud. The proposed solution (called DMCS^3) will provided an hardened cloud storage environment that will be resistant to user's data breaches either by breaking into a user's account or by compromising third party service providers, due to its multi-cloud system implementation, its authentication process and BDLT integration.

The proposed solution is suitable both for private and public usage, and its implementation can be made recurring both to public, private or hybrid CSPs, however, and due to it's BDLT recording process, both the intended usage and implementation of this system has to be made according to the type of blockchain selected. The criteria previously described in figure 2.13 will be of utmost importance and, combining the aforementioned flowchart with the existing, or planned, structure one has related to CSPs, and in order to meet the intended security goals for our needs, a feasible solution should be achievable if all the criteria mentioned below in section 3.2 is met and correctly implemented.

3.1 Solution Objectives and Requirements

Based on the goals that were previously mentioned, three distinct objectives were defined so that a feasible and secure solution design could be specified using BDLT while achieving full service availability due to its distributed storage system. The proposed solution design, detailed in section 3.2, aims to provide:

1. **Security, Traceability and Auditability**(section 3.1.1)
2. **Availability**(section 3.1.2)
3. **Resilience**(section 3.1.3)

In order for the solution to achieve the intended objectives, several requirements must be previously met and from which we can split into:

1. **System Requirements**:
   (a) The system has to be deployed on a totally decentralized basis
   (b) The registration and authentication processes must be recorded in a blockchain
   (c) The Multi-Dimensional hash matrix (section 3.2.7) has to be recorded in a blockchain
   (d) The file storage has to be made recurring to several CSPs
   (e) After the assembly file (section 3.2.5) and the original file's fragments and its copies are uploaded into the CSPs, the original file must be automatically deleted from the device
   (f) The file's fragments must be uploaded in a shared folder (section 3.2.6 replicted into the several CSPs
   (g) Redundancy has to be assured both in files and in CSPs (section 3.2.4)
2. **Security and Compliance Requirements**:
   (a) The system will limit access to authorised and registered users only

(b) It will only be possible to log in with a registered ID Pair (section 3.2.3)

(c) One Device ID (section 3.2.2) can only be associated to one single User ID (section 3.2.1)

(d) One User ID can have multiple Device IDs associated to it

(e) Every User ID must be unique and must be recorded in a blockchain

(f) Every Device ID must be unique and must be recorded in a blockchain

(g) Every ID Pair must be unique and must be recorded in a blockchain

(h) Every Shared Folder ID must be unique and must be recorded in a blockchain

(i) It will only be possible to share files with and between registered ID Pairs

(j) There must be a shared folder between specific users, with a specific and Unique ID recorded in a blockchain

(k) Every assembly file must be encrypted with the owner's private key first and with the other user's public key afterwards

(l) The file's fragments must be also encrypted and spread into different CSPs

3.1.1 Security, Traceability and Auditability

Aiming to achieve increased security, traceability and auditability, the solution presented in this work will combine state-of-the-art techniques, such as BDLT, cloud computing and distributed systems in an harmonised way, allowing users and organisations to store their critical data remotely without the need for a centralised management entity, due to its fully decentralized approaches both in registration and authentication processes and in data storage and sharing processes. Users, and only them and the people allowed by them, will have access to their own personal data in a secure way if all the prerequisites listed above and detailed below in the following sections are achieved, meaning that means will be provided

for users to have full control of their data, minimising data breaches and leakages either on their own devices or in third party service providers due to its data and CSP redundancy approach.

3.1.2 Availability

Without compromising the previously mentioned objective, this work also aims to provide the means for a service that ensures high availability of data, by recurring to data redundancy and duplication and spreading it across several CSPs in the network. With this type of implementation, not only data backups are assured but service backups as well, meaning that if a certain CSP on which the solution relies on is temporarily down or unavailable, the solution will still be up and running and the users will still have access to their data which, ultimately, it's every user's prerequisite when they rely on any data storage solution; to have full access to all their data, on-demand.

3.1.3 Resilience

On top of the previous two objectives, the proposed solution also aims to mitigate cyber threats and fight cyber attacks who aim to steal personal data or personal files, or suppress the access to those, in a very efficient way, without compromising either the user's or their device's details, and without compromising their cloud stored files. Resilience is, therefore, subject of great concern for users since it will ultimately dictate how the solution can increase it's cyber attack's and cyber threat's resistance. The aforementioned objective should be achievable solemnly and automatically due to the solution's features and architecture specified in the next sections, and without the need of intervention from the user.

3.2 Proposed Solution Design

In order to meet the objectives identified, the proposed design for this solution implements and combines Public-Key Infrastructures (PKIs) , a Multi-Dimensional hash matrix, BDLT, cloud computing and distributed storage systems so that an efficient and effectively secure system is achieved based on the requirements listed at the beginning of the present chapter.

Therefore, and to better detail the solution and how it will operate to fulfil the system objectives, 8 different processes and 1 sub-process, done automatically once the first 2 processes are complete, will be considered and detailed:

1. **Account creation process**: the user creates his account based on blockchain crypto-wallets principles;
2. **Device authentication process**: will detail how a device will be uniquely identified within this system and assigned solemnly to a single user, generating the **ID Pair**, which is the combination of both the user id and one device id;
3. **File splitting and storage process**: describes in detail how the file is split and how both the fragments and the record of that process is stored, and when, in order to be possible to re-assemble the file anytime when needed;
4. **Assembly file generation and file sharing processes**: describe in detail how an assembly file will be generated and later on how it will be shared with third parties without compromising its integrity and availability, despite none of the entities involved (both users and/or devices) being in possession of the complete, original file at any given moment in time, except when accessing it;
5. **Multi-Dimensional hash matrix**, which is authenticated in the blockchain and will contain the pairing information of users, devices, and shared folders between them;

6. **Original document access process**, which describes how and when a specific user has access to a shared file by means of a shared folder;

7. **Public Ledger and Blockchain Recording process**: will describe which transactions will be recorded and when, on the blockchain.

The first 2 processes listed above, and which will later generate the ID Pair, are depicted below in figure 3.1 and are detailed in sections 3.2.1, 3.2.2 and 3.2.3.

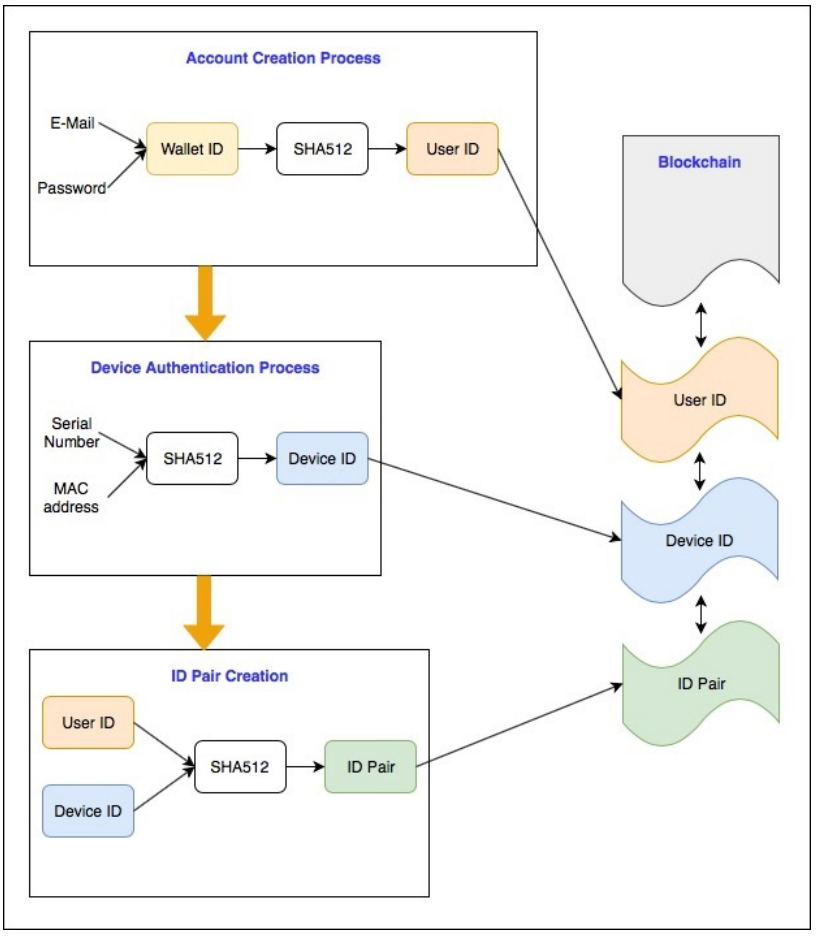

FIGURE 3.1: ID Pair creation sub-process

The desired outputs, which are previously specified in all the solution objectives, are achievable due to a function call within the application, after all the authentication criteria is met. When an authorized user tries to access an original file or a legit copy of it, the assembly file is read inside the app, calling all the fragments from multiple CSPs and after decrypting them, assembles all the fragments

to build the original file. This output is expected to be successfully achievable if all the processes listed above and detailed below are fully met, being each one of them of equal importance for the proposed solution design.

3.2.1 Account creation

When creating an account for the first time, a user will need to have an associated e-mail to his account and will also have to define a password. After submitting the two in the solution's application or website, he will then receive an e-mail with his Wallet ID (W_{α}^{ID}), which is a unique identifier composed by 32 alphanumeric characters and 4 dashes in between, for example "8a34pa4d-3d5c-6745-d282-bc974d64pcf2", and a hash value of this ID, which will be referred from now on as the User ID (U_{x}^{ID}) = $SHA512[W_{\alpha}^{ID}]$, $\forall x, \alpha \in \mathbb{N}_{\neq 0}$, will then be generated and stored in a Multi-Dimensional hash matrix which will later on be detailed.

This hash value will be calculated using to the SHA512 algorithm because even though there are no successfully demonstrated attacks on a SHA256 algorithm, some authors were able to demonstrate in their work "hash function collisions on 31 steps of SHA256 with complexity $2^{65.5}$, and semi-free-start collisions on 38 steps with complexity 2^{37}" [72], and therefore, the chosen algorithm would not be this one but SHA512 instead which, even though it takes slighly longer time to calculate, it is proven to be more secure with a collision resistance claim of 2^{256} [73] and a preimage resistance claim of 2^{512} [74].

While at the time this U_{x}^{ID} value is calculated and stored in the aforementioned Multi-Dimensional matrix (section 3.2.7), both the W_{α}^{ID} value and the user's hashed password value will also be stored in the blockchain, so that every time a user wants to log in into his account/wallet and enters the password, the hashed value of it will be compared and if there is a match he then is able to successfully log in.

3.2.2 Device authentication

After a user account is created, at least one device should be associated with it in order for a user to be able to use this solution. When authenticating the device for the first time, the application would retrieve both the device's serial number (S^{β}) and it's MAC address (M^{ϕ}) so that a unique identifier of this device can be generated and, despite the device's MAC address is either in a 48 or 64-bit basis, a conversion of it would be made into hexadecimal notation. This hexadecimal number will later work as a salt when hashing the device's serial number recurring to a SHA512 algorithm, returning the Device ID $(D_y^{ID}) = SHA512[(S^{\beta}) + (M^{\phi})]$, $\forall y, \beta, \phi \in \mathbb{N}_{\neq 0}$.

The choice for a SHA512 algorithm in specific, instead of other hashing algorithms, happens for the reasons mention above in the account creation process, however, a salt is also used in order to decrease possible attack and corruption chances since once there is a unique salt for every device, an attack has to be made individually and targeted to a user's specific device. Once the salted hash value is calculated, it is then stored in the blockchain with its associated timestamp and device owner, or U_x^{ID}, being all the computational work made automatically within the application every time a user logs in into his account in a specific device. Therefore, there is no need to store the salt anywhere, reducing even more the chances of corruption of the authentication process.

To the combination of both the U_x^{ID} and the D_y^{ID}, defined by this salted Hash, is the so called ID Pair which will be detailed in the next section.

3.2.3 ID Pair

The from now on defined ID Pair (P_k^{ID}) is a unique ID defined by the combination of a U_x^{ID} and a D_y^{ID}, both previously defined and detailed in sections 3.2.1 and 3.2.2 respectively.

This P_k^{ID} will work as a reference within a file assignment matrix, the so called "Multi-Dimensional Hash Matrix" detailed in section 3.2.7 below, and it will be uniquely identified by a salted hash composed by the U_x^{ID} and D_y^{ID} in a way that $P_k^{ID} = SHA512[(U_x^{ID}) + (D_y^{ID})]$, $\forall k, x, y \in \mathbb{N}_{\neq 0}$.

After computed, the P_k^{ID} will then be used to assign assembly files to different users and their devices by means of a shared folder, and therefore its importance. A practical example of how this distinction can be important is if a user wants to share a classified document from its company with another user, and this document must remain accessible only in the companies assigned devices. A user might use both is company issued laptop and cellphone for work-related purposes, but his cellphone is more likely to be used for personal use as well, therefore, while when sharing a document with another user, and if the intended purpose is solemnly work-related, the most desired destination would be this specific user's company-issued laptop, and not all his devices, in order to mitigate undesired information leakage and/or propagation.

Both the assembly files and the shared folder mentioned in the previous paragraph will later be described in sections 3.2.5 and 3.2.6 respectively.

3.2.4 File splitting and storage

When storing a file in multiple CSPs, it is important to emphasize that this solution is meant to store not a single file and several copies of it in several cloud services but dozens or even hundreds of fragments of the original file, depending on its size, plus redundancy copies of those fragments, in a diverse variety of CSPs, both private, public or hybrid, depending of the intended purpose or pre-requisites defined by everyone who intends to rely on this proposed solution. After a file is uploaded into the application, several redundancy copies will be made and all of these files will be automatically split in similar size fragments, which will later be encrypted with the file owner's private key, in a similar process as the one described in [75].

After this process, a hash of every encrypted fragment will be generated and the fragments itself will be dispatched to different CSPs in a random order. The destination of every fragment will be recorded in the assembly file, and the assembly order of those fragments will be given by a timestamp, generated at the time of the fragment's encryption process, which will be added as a flag at the beginning of the fragment's hash, so that when reading the assembly file instructions an accurate and precise re-assembly process can be achieved, even if remotely in other devices by other, authorized users.

As detailed and defended in [76], file allocation and its several redundancy copies in multiple systems should depend, at least, on the intended storage costs, transmission costs, file length, and the storage capacity of each computer, in this case, cloud server. Since cloud storage capacity is being outsourced and it is virtually unlimited, we would take away this variable out of the equation, meaning that a file's redundancy copies would depend on the other factors, nevertheless, if we take into account RAID's approach [77], and in this case, RAID level 6, that implements a 3 disk redundancy which guarantees a 0,03% probability of data loss with 38052 years for mean time to data loss, and transpose it to this solution, at least 3 CSPs should be used and 3 redundancy copies should be made in order to guarantee an extremely high data availability.

After this file splitting and allocation process is complete, and since the original file and several redundancy copies are already stored in different CSPs, there is no need for the file to remain stored also in the user's device and therefore, a physical deletion of this file would then occur, assuring that undesired discloses of the file would not be possible by breaking into the user's device.

This aforementioned assembly file, detailed below in section 3.2.5, is the actual file which will be used for sharing and granting access to the original document between users in the process described in section 3.2.6.

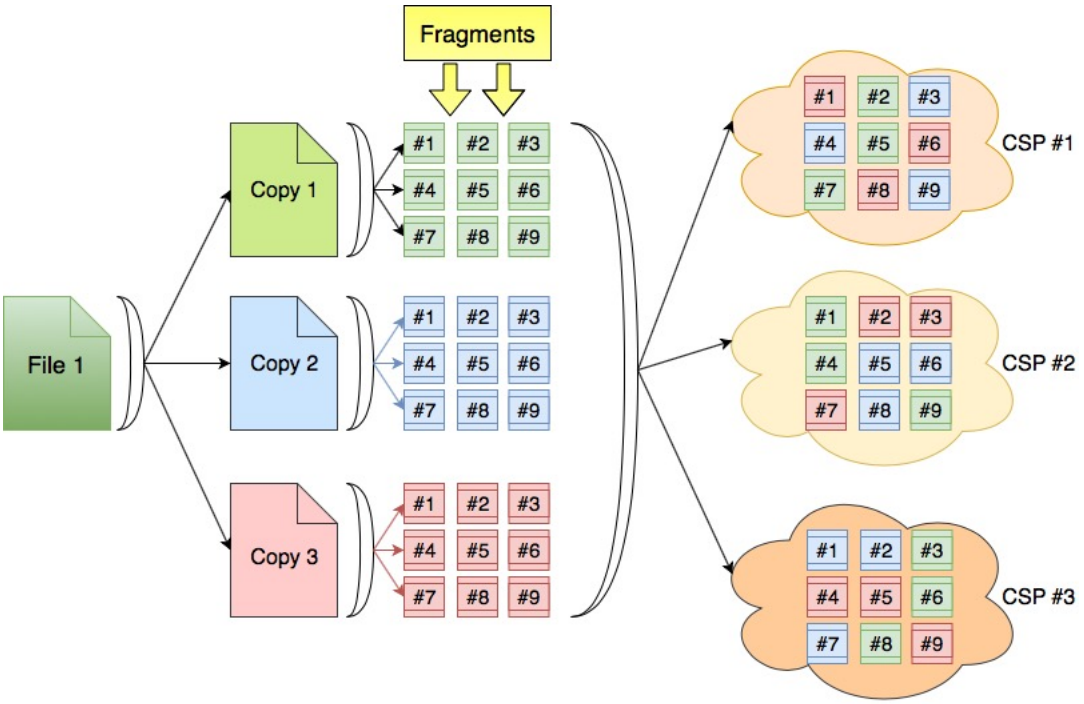

FIGURE 3.2: File Splitting and Storage

3.2.5 Assembly file

The assembly file is a text file the application will read in order to know the assembly instructions for a specific file, allowing every user who has access to it to access an original file by means of a shared folder, which will be detailed next in section 3.2.6. This text file will contain all the information needed to re-assemble the original file, even if by other users than the owner of the original file itself.

Starting from the description of the original file, the associated creation time in the application, given by a timestamp, and its owner, so that in the application's user interface an automatic association can be made without having the need to re-assemble every file in order to find the intended one, all the relevant fragment's information, both from the original file and its copies, will be contained in it.

The relevant information associated with every fragment will consist in a string, for each of them, given by its assembly order which is returned by a timestamp at the time the fragment is created when splitting the original file and its copies, the hash value of the encrypted fragment, and finally the destination CSP IP address

range. Since all the CSPs operate in a wide range of IP addresses instead of only one, and knowing that some times our data can migrate from one server to another, with a different iP address, it is not feasible assigning a single IP address to a fragment.

Also, and for control purposes, the assembly file will contain both the total size of the original file, or a copy of it, in bits, plus every fragment's size, both encrypted and decrypted, also in bits.

3.2.6 Assembly file sharing

Every time a user grants access to other users to any of his files, a communication channel is opened by means of a shared folder between them and, along with this folder, an ID of it (F_z^{ID}) is also generated by means of the hashed combination of every P_k^{ID}, being $F_z^{ID} = SHA512[(P_1^{ID}) + (P_2^{ID}) + (P_{...}^{ID}) + (P_k^{ID})]$, $\forall k, z \in \mathbb{N}_{\neq 0}$, which is also stored in the Multi-Dimensional hash matrix.

In order to increase availability, access speed, and to fight possible undesired downtimes in third party CSPs, redundancy copies of these folders will be created in an order that the total amount of shared folders will be the exact same as the total amount of different CSPs used by the total amount of users using a particular shared folder. Nonetheless, and while sharing a file with one or more users, the owner of the file only has to share it once with the intended users; the process should be repeated automatically as many times as there are shared folders between the users.

When adding assembly files to these folders, every owner, and only the legit owner, has the ability to move these files in and out of the shared folder, and when they do so, every other user who has access to the shared folder will no longer be able to see the moved assembly files. Since this shared folder works as a file communication channel between 2 or more different users, every user who has access to it can add or remove his own files, meaning that there is no need to open a new communication channel (i.e., a new folder) for every file to

be shared between the exact same users. The shared folder and its redundancy copies will only be available to every user who has access to them, being this access authorization checked on a Multi-Dimensional hash matrix like the one described below in section 3.2.7.

It is also of utmost importance to emphasize than when sharing an assembly file with other users, a copy of it will be made for every user who has access to the shared folder and every copy will be encrypted first with the owner's private key and with the other user(s) public key(s) afterwards in order to assure both authentication and non-repudiation of every file, while keeping full auditability of every access and granting it only to the targeted and desired individuals.

On the other hand, if the owner of a file wants to revoke the access he previously gave to his files, either to a single user or a group of users, he then has to remove the shared assembly file from the folder shared with the intended user(s). While doing this, the copies present in the redundancy shared folders will also be deleted in the same way they were created and, if the deleted file was the only on this shared folder, than this F_z^{ID} will be removed from the Multi-Dimensional hash matrix and a record of this action will also be stored in the blockchain.

3.2.7 Multi-Dimensional Hash Matrix

The Multi-Dimensional hash matrix referred in this work is aimed to better address, in an unambiguous way, how both users, user devices and shared folders between specific user's devices will be referenced to each other.

To better picture how this matrix would work, we will start with the U_x^{ID} and the D_y^{ID}, on which the combination of both will result in the P_k^{ID} that identifies unambiguously a specific device assigned to one and only one specific user like previously explained in section 3.2.3. If one single user has only one device associate to his account, then we would have a $P_{1,1}^{ID}$ matrix on which the result would be the concatenation of both the U_x^{ID} and D_y^{ID} values, i.e., the P_k^{ID}. If the user, however, has Y devices, then we would have a $P_k^{ID} = P_{1,y}^{ID}$ matrix, resulting in

$P_k^{ID} = P_{1,y}^{ID} = \begin{pmatrix} P_{1,1}^{ID} & P_{1,2}^{ID} & \cdots & P_{1,y}^{ID} \end{pmatrix}$ and therefore meaning that the amount of ID Pairs (P_k^{ID}) would correspond to the same amount of different values of Y. On the other hand, having X users with Y devices each, this would result in a $P_{x,y}^{ID}$ matrix, like the one depicted below, on which each user has its own specific amount of devices, not being required, however, for every user to have the same number of devices as the others.

$$P_{x,y}^{ID} = \begin{pmatrix} P_{1,1}^{ID} & P_{1,2}^{ID} & \cdots & P_{1,y}^{ID} \\ P_{2,1}^{ID} & P_{2,2}^{ID} & \cdots & P_{2,y}^{ID} \\ \vdots & \vdots & \ddots & \vdots \\ P_{x,1}^{ID} & P_{x,2}^{ID} & \cdots & P_{x,y}^{ID} \end{pmatrix}$$

Having the assignment matrix concluded, and assuming that for every different value of $P_{x,y}^{ID}$ there is also a different P_k^{ID}, i.e. $P_{x,y}^{ID} = P_k^{ID}$, a pairing vector should also be included so that we can associate a P_k^{ID} granting access to his files to another P_k^{ID} receiving access to other users P_k^{ID} files, and therefore, by adding a "Z" vector to the existing matrix with U_x^{ID}, we would end up with 2 U_x^{ID} vectors and 1 D_y^{ID} vector.

The concatenation of both $P_{x,y}^{ID}$ values will result in the shared folder ID (F_z^{ID}) and its redundancy copies, serving as a referral to the actual folders stored in the several CSPs in use, and if, however, a file is to be shared with multiple users on multiple devices at the same time, then the shared folder ID (F_z^{ID}) would be comprised by the concatenation of all the P_k^{ID} values.

However, since this matrix won't exist as a physical matrix itself, but rather as a virtual one like previously mentioned, stored on the blockchain, its integrity, availability and non-repudiation are assured, while helping to maintain both user IDs and their data confidential, always available and firmly secured.

3.2.8 Original document reading access

After an assembly file is shared, the authorized users will be able to access it in the shared folder by means of multiple authentication factors. The first one starts

when the user logs in into his account with his password and, after logging in, it will be automatically checked by the application if the device the user is using is one of the devices associated with his account. If not, the log in is not successful and the application requests the user to log in in an authorized device. If both the user password is correct and the device is authenticated and associated with this user, all the shared folders between other users and this user/device pair will show up available on the user interface, with all the shared assembly files inside it.

This process described above is achievable by means of the Multi-Dimensional Hash matrix described above in section 3.2.7.

The second authentication process, concerning the access to a specific shared file, will occur when the user selects it on the user interface. The 2-way public-key decryption takes place and the assembly file should now be accessed. After accessing the intended assembly file, the application reads it and starts the execution process by calling all the CSPs for all the fragments, comparing their hash values. If the values match, then the fragments will be collected in the device and the assembling process will start in the order defined by the timestamp given when splitting and encrypting the original file and its copies, starting with its decryption with the owner's public key. After the decryption process is complete, all the fragments are accordingly ordered and re-assembled, and the user who was granted access can now have access to the original file or a redundancy copy of it

It is, however, of utmost relevance to mention that every user can have multiple devices associated with his account, and each user-device pair may have different shared folders associated. This way, and recurring to this Multi-Dimensional hash matrix, it is assured that every "user/device" pair will only be able to access the assembly files shared with it inside the shared folder, and since only the legit owner of those assembly files has the permissions to grant or revoke access to a specific file, the user who is accessing it can not share it with other user-device pairs. Nevertheless, if somehow a user who is granted access to a specific assembly file is able to corrupt the application and re-send it, it is relevant to mention that

the writing process in this Multi-Dimensional hash matrix happens automatically when the legit owner of the file shares it or revokes access to it in the shared folder between them, meaning that if the application is corrupted and someone is able to re-send a file to someone or someone else's device, there has to be a shared folder between those user-device pairs as well, this folder has to be recorded in the Blockchain with the correct timestamp, the hash value of this file must be associated with this shared folder's hash value , and the user who is re-sending it has to disclose his private key to third party users since every shared file is encrypted in a 2-way process as described above in section 3.2.6.

3.2.9 Public ledger and Blockchain recording

On top of the process mentioned above in section 3.2.8, several processes and steps are recorded in a blockchain so that a trustworthy solution is achieved, starting when a user creates an account for the first time. This user will have an identity (ID) associated to a hash value stored in the Multi-Dimension matrix, and this ID will be recorded in a blockchain so that by the moment the user is created there is a record of him with an associated timestamp in a blockchain on a process identical to the one depicted in figure 2.11.

Also every user's devices will be added to the blockchain in a process like the one described before by the time they are added to the application and associated to a specific user. This way, and since every record of every device will have a timestamp associated to it, generated at the time the user authenticates his device inside the application, it no longer can be dissociated from the owner on the blockchain, as long as the owner doesn't dissociate it on purpose. If so, this deletion process will also be recorded on the blockchain, meaning that from that moment on the device in question can be added by a different user.

When a user grants access to his files to other registered users, a record of it will also be stored on the blockchain in the same way as the process depicted in figure 2.10, meaning that if the access is granted to multiple users, this process will

be repeated as many times as there are users who is granted access. This way, and by means of the achieved consensus on the blockchain, it is unquestionable that the users exist, the file exists, and the access was given to someone at any given time and, if the access is later revoked, the action is also stored on the blockchain. Nevertheless, it is of utmost importance to emphasize the fact that this record is for the file access grant, and not for the existing communication channel between the users. For this folder on the other hand, a new record will be also generated in the blockchain when it is created, with its associated timestamp, and once this communication channel, or the shared folder, is created, it will not be possible to delete it at any given time unless all of the user-device pairs associated to it are also deleted. In that case, the deletion will be automatic when there is only one remaining user of that shared folder, and a record of this deletion will also be stored in the blockchain with the associated timestamp.

With this semi-complex method, and due to its distributed implementation, a transparency and authenticity increase will be perceived by its users since everything that is recorded in the blockchain is immutable and all the related data is kept complete, accurate, consistent, timely and widely available and, like in almost all the processes described above, also every blockchain recording should be done automatically by the application without the need for the user to start the process or having the ability to pause it or revoke it at any time.

Chapter 4

Solution's Security Assessment

In order to better understand if our initial objectives defined in section 3.1 are fully achieved, several security characteristics were considered and detailed below so that the following hypothesis could be verified:

1. **There is increased security, traceability or auditability with the implementation of the aforementioned proposed solution**
2. **The proposed solution provides full service availability**
3. **The proposed solution demonstrates increased resilience features when compared to other available solutions**

In the present chapter, and for each one of these hypothesis, a thorough research will be made regarding other available and similar solutions and a confirmation or rejection of them will be considered based on several sub-topics within the scope of each one of the points considered above in section 3.2 regarding the proposed solution design. Also, for the first and third objectives mentioned in section 3.1 a comparison matrix and an adversary model will be, respectively, built, and regarding the solutions requirements, a peer comparison will be made with solutions already available to the the public that offer similar capabilities, functionalities and services as the ones presented by this solution.

4.1 Solution's security characteristics and self-assessment

Being one of the core objectives of the proposed solution to present a reliable, efficient, very secure and auditable solution, the present section will study and make a theoretical assessment of the presented work in respect to increased and enhanced security, traceability and auditability. For this purpose, an assessment will be made first regarding its Multi-Dimensional Hash-Matrix Authentication, followed by the file splitting and multi-cloud storage feature and, finally, the blockchain and the records that are kept in it.

4.1.1 Multi-Dimensional Hash Matrix Authentication

In order to be able to use this solution a user has to be registered and needs to have at least one device associated to his User ID, so that an ID Pair can exist in the matrix. If there is no ID Pair whatsoever it means that either nor a user and an associated device are registered within the blockchain or, if in fact a user is registered he or she doesn't have a device associated to his/her account and therefore it will not be possible to send or receive files to and from other users. This happens because, like explained above in sections 3.2.6 and 3.2.7, the assembly files are shared inside shared folders on which the ID is created by means of the hashed combination of every involved user's ID.

Due to its complexity and integration with BDLT, every ID in use is very hard to tamper with since there has to be an achieved consensus within the nodes for both the account and device ID's creation and their authentication every time a user wants to use the proposed solution, and therefore, due to blockchain's characteristics, the matrix would be theoretically impossible to tamper with due to every ID's chain of events and dependencies, plus the computational complexity involved to either gain at least 50% of all the computational power involved in order to get the blockchain's control or, instead, to alter every record from the time that specific user account was created until the present time which is, with today's technology, impossible to do.

Also, and since not only a device is uniquely assigned to a user, returning a unique ID Pair for each user/device pair in use, the shared folder's ID is made by the hashed combination of these user's ID pairs, meaning that, theoretically, if someone wants to gain access to a specific shared folder, the easiest way, if not by having physical access to the device and knowing the owner's private key, would be by knowing the original file owner's wallet ID, his private key, and the specific devices serial numbers and MAC addresses, so then the hashed combinations could be generated in order to get the intended access, by emulating these values and fooling the system. It is, very likely, the only possible way to override the system, involving several dozens of characters from which not even the actual legit user knows by memory, except from his private key.

Nevertheless, this system offers the means to uniquely identify who has a legit access to specific files, and on which devices, meaning that a very granular filtration can be achieved and assuring that even if a user is granted access to a file from and to a specific device, he will not be authorized or even able to either move it or re-send it to other people, even to himself, on another device. Since the file does not belong to him, he shouldn't have the permissions to do whatever he wants with it, even if shared with him, because that would be considered an abuse of trust and therefore, the system would also mitigate these types of abuses by eliminating them by default thanks to this multi-dimensional hash matrix and its file assignment rules.

4.1.2 File Splitting and Multi-Cloud Storage

Due to the solution's file splitting and multi-cloud storage features, it is assured that even if there is a security breach within a CSP the attackers wont be able to retrieve any usefull data from it. Firstly, because there won't be full, assembled files, in any CSP. There will only be fragments of the original file's copies and all the assembly files, with a 2-way public-key encryption like previously described, and shared folders with an ID associated to them that are impossible to trace back to any user in the network.

In other words, even if any CSP is hacked and all the data in it is retrieved, the attackers would either need to trace back a specific folder to specific users by means of trial and error with their combined hashed values so they could have access to a specific file (and then they would also need both the user's public and private keys), or they would just try to re-assemble every fragment within the CSP in order to achieve the file they wanted - which is very likely impossible in the 2 scenarios.

This way, it is almost assured and guaranteed that a direct access to data by means of a CSP breach/theft is virtually and computationally impossible with today's available technology, meaning that the only, theoretical way to access a file would be through the authentication process, either by surpassing the multi-dimensional hash matrix access criteria, by altering and/or tamper blockchain's records, or through a (seemingly) legit access.

4.1.3 Blockchain Records

The blockchain records are unquestionably relevant to both the authentication processes since by the time every ID is created there is a timestamp to record it and therefore not only exists a record of that ID that will remain forever untampered, but there's also an immutable registration of the time that ID was created. This way, everything can be uniquely identified without the worries of some records being eventually modified in the future in an unlegitimate way. From the time a user ID is created with a wallet ID, a device ID is generated and an ID pair is associated with the user and his device ID's, everything is guaranteed to follow this specific order of events in the blockchain. Also, for the shared folder's ID, there also has to be the record of the involved user's ID pairs, meaning that if they dont exist in the blockchain, the process is not fit to continue and it will cease. Since there is always a record of everything associated to any user in the blockchain, full traceability can be achieved and it is granted with this process that only the authorized people will have a legit access to a specific file, and therefore,

all accesses have audit capabilities and can be easily traced to specific users in specific devices at any given moment in time.

If there is a file disclosure, there are only 3 possibilities: either a certain CSP was breached and every file is corrupted if the attackers got a way to surpass every security mechanism (which, even though not 100% impossible, is theoretically extremely difficult to happen), an authorized user who had access to it disclosed it, or someone who happens to know an authorized user's credentials and has physical access to his devices disclosed it. In both these last 2 cases, probably the most likely to happen, the disclosure can be easily traced to someone.

4.2 Attack Scenarios and Adversary Model

Nowadays, the costs and repercussions of a successful cyber attack are mind-boggling, specially when we look at the numbers provided both by Microsoft, Juniper Research and Symantec. If, from one side, Microsoft is certain that during 2018 a company's average cost of a data breach would settle at around $3,8 Million [78], Juniper Research says that this number will rise almost 4000% to $150 Million by the end of 2020 [79]. Data is, without a doubt, one of the biggest assets people and businesses have at their disposal, so it comes whit no surprise the fact that during last year ransomware attacks had an increase of 36% when compared to 2016[80], with a tendency to increase at a substantial pace.

The concerns involved in worldwide cyber attacks aimed to disrupt businesses and services, steal and make data unavailable, or with other equally bad motives are in such a way so staggering that Warren Buffett recently stated that cyber attacks are "the number one problem with mankind" [81], and therefore, the importance of studying and comparing this solution with current solutions available in the market regarding cyber resilience and resistence.

Some authors [82] demonstrate and share similar concerns and vision as the one presented in this work, while defending that data should be encrypted and

packed with a usage policy, while being only visible in a trustworthy environment. This work aims in the same direction in order to mitigate the associated risks of an IoT-driven society like mentioned above and corroborated by several entities, and therefore, in the present section a theoretical comparison study between the presented solution and solutions available nowadays will be conducted in order to assess and prove or disprove if there is no increased cyber attack's resistance when they all are equally compared within the same attack scenarios.

Knowing if our data and files remain private and timely available is the subject of study in this section when the solution is confronted with DDoS, Ransomware, Brute Force and Phishing attacks. Even though only 4, and not all, types of attacks will be considered, a self-assessment will be made and a comparison with other known and available solutions will later be addressed in section 4.3 in order to assess if the presented solution offers an increased cyber attack resistance in these 4 cases.

For this purpose, an adversary model will be made for the 4 different scenarios with the following attributes:

1. **Adversary type** - AT
2. **Campaign objective** - CO
3. **Campaign vehicle** - CV
4. **Campaign weapon** - CW
5. **Payload delivery** - PD
6. **Payload capabilities** - PC

1. **Scenario 1 - DDoS Attack**
   (a) **AT** - Commercial Hacking
   (b) **CO** - DDoS
   (c) **CV** - Botnet

(d) **CW** - User-installed malware

(e) **PD** - Process hijacking

(f) **PC** - DDoS

2. **Scenario 2 - Ransomware Attack**

(a) **AT** - Hacktivist

(b) **CO** - Data control for extortion

(c) **CV** - Phishing with link/attachment

(d) **CW** - User-installed malware

(e) **PD** - Executable file

(f) **PC** - Ransomware

3. **Scenario 3 - Brute Force Attack**

(a) **AT** - Hacktivist

(b) **CO** - Account takeover

(c) **CV** - Remote login

(d) **CW** - Socially engineered remote access

(e) **PD** - Scripting

(f) **PC** - Data exfiltration; Backdoor for remote access; Privilege escalation

4. **Scenario 4 - Phishing Attack**

(a) **AT** - Hacktivist

(b) **CO** - Intellectual property theft

(c) **CV** - Spear-phish with link/attachment

(d) **CW** - User-installed malware

(e) **PD** - Executable file

(f) **PC** - Data exfiltration; Backdoor for remote access; Privilege escalation

4.2.0.1 DDoS Attacks

A DDoS attack consists of a tentative disruption of a service by flooding it with requests, or packets, from multiple sources at the same time. Usually, and according to Mirkovic and Reiher [83], some of the most frequent motives of such an attack are personal reasons and material gain and, nowadays, these attacks got so evolved they are capable of "carrying out several functions at once", like shutting down a firewall or steal data [84]. Also, in some cases like the one mentioned previously in section 2.1.1, and as defended by the author, "the true victim might not be the actual target, but others who rely on the target's correct operation" [83], like third party CSPs, and therefore its study importance on this solution. Being able to be immune to such an attack has great implications both in private lifes and in businesses, as it means the data is always timely available no matter what, and with the increased impact data is having in our everyday life, it becomes an asset being able to use that data whenever needed, specially when decision making opportunities rely on it.

The above presented solution presents a crucial asset that mitigate a data access interruption by means, for example, of a DDoS attack, and its it's multi-cloud architecture. By dividing data through multiple CSPs, with redundancy copies spread across all service providers, even if one CSP, or two, or three, suffer a simultaneous DDoS attack, the above presented solution relies on the other CSPs to make data timely available to its users. It is, of course, the user's decision on how many CSPs he will develop his architecture in, since its chosen type of cloud storage might be public, private or hybrid and therefore the importance of accessing what is really needed for him or his business. Relying on a public or on a Hybrid architecture would imply relying on third party CSPs, on which some already have DDoS mitigation tools to rely on, like AWS Shield or Google Cloud's Cloud Armor, and therefore, this architectural system presented on this solution would serve, in these cases, like a second line of defense in case a DDoS attack is, in fact successful, being equally important and also attack resilient in a case where a third party CSP does not rely on a tool that is aimed to prevent such attacks.

Overall, and when compared to single cloud solutions like AWS, Google Docs or Dropbox, it presents increased data availability capabilities when confronted with a successful DDoS attack aimed to the solution itself. On the other hand, if compared with similar, multi-cloud solutions, data availability in such cases would be similar, since the same principles would apply to other multi-cloud architecture solutions like Storj.

4.2.0.2 Ransomware Attacks

These types of attacks first appeared in 1989 with the AIDS Trojan, but it was in 2017 when they got more notorious whit what was considered the worst ransomware attack ever, when WannaCry disrupted companie's, hospital's and government agency's work worldwide [85]. Such an attack consists of a piece of malware that hijacks user's data by encrypting it, demanding a payment, or ransom, in exchange for the decryption key [86], and according to Brewer [87], the FBI estimated incurred business losses of roughly 1 Billion US$ during the year 2016 while stating that everyday more than 4000 attacks like these occur worldwide [88]. Due to its characteristics, namely global and fully anonymous, typically these payments are made with crypto currencies like bitcoin, who guarantee anonimity throughout the network [87], meaning that the recipient's identity remains anonymous to everyone, including the authorities, and therefore the provenance and identity of the attackers is also kept undisclosed.

By hijacking user's files in their own devices, or in third party CSPs, which was never seen so far, ransomware attacks represent a tremendous threat to businesses and individuals who are not keen either to loose their data forever or having to pay to have it back, and through the architecture presented in this work it can easily be demonstrated that such attacks do not actually represent a serious threat to the solution's users if they share the access, at least, within their own device environment, i.e., with every device they have. By relying on a system that splits original files and its redundancy copies, encrypts those fragments, and then spreads them throughout several CSPs, deleting the original file from the device itself, the

only possible way to encrypt every original file would be to inject the malware and hijack all the data in all the CSPs available, which even though it is possible, it is not very likely nowadays. Nonetheless, there can be the case that a device is infected with this type of malware, and even if original files are not corrupted then the assembly file would, meaning that if a user does not share it within it's own device environment, then the access to that file would be permanently lost to him but not to the other people he shared access with. Nevertheless, if a user wants the same type of data portability within his devices like he has with CSPs like Dropbox, Google Docs and others, then he would have to individually grant that access, meaning that a similar protection would be offered since the file would still be available even though some of the devices were infected and corrupted.

Even though not as user friendly as similar cloud storage services, the presented solution is one of the few capable of preventing data disruption or unavailability if an attack occurs on third party CSPs due to its distributed, multi-cloud architecture. If the attack, however, is successful and the user does not share the assembly file whatsoever, either within its device environment or with other people, then the access to that file would be permanently lost and he wouldn't be able to get it back unless he payed the ransom, meaning that a more traditional approach to a third party CSP would pay back since there is no data inside the user's device whatsoever and he will never lose the access to it.

4.2.0.3 Brute Force Attacks

A Brute force attack is a password guessing technique that consists of a systematic attempt to defeat an authentication mechanism using a trial and error approach [89] [90]. There are several tools to perpetrate such attacks, like "Hashcat", "John the Ripper" or "RainbowCrack", as well as several techniques being one of the most common based on pre-compiled lists, the so-called "dictionaries", like demonstrated in [91], or even a special kind of dictionary, the rainbow table [92], which is a pre-computed list of hash values specific to a certain hash function. These dictionaries are meant to help the cracking process and can be based both

on lists made available on the web from cracked accounts, websites and servers, or in custom-made lists generated by online or custom-made scripts based on inputs and data we know about a specific target. By comparing both hashes, from the password in the dictionary and the one stored in the service's database, concerning the hashed value of the actual password, the attack is either successful or not if there is a match between those two hash values.

These types of attacks fall into the offline attacks category, since the attack is done in our own system or, alternatively, in the victim's system if we have local access to it, meaning there is no need for a network connection with the victim's system or device. While for online brute force attacks there are available several mitigation tools and techniques, like blocking an account after a few failed logins, Multi-Factor Authentication (MFA) or limiting the number of login attempts, some authors state that salting password hashes would defeat any offline brute force attack [93], making it a valuable insight for this work. In fact, and despite the fact online brute force attack mitigation tools are not considered in the design of this solution, it is of utmost importance to emphasize that salted hash passwords are considered both for user and device authentication, resulting in another salted hash password, for the ID Pair, with a SHA512 algorithm, that would, in theory, make the proposed solution resistant to any offline brute force attack.

As to on what online brute force attacks concerns, the solution would be, in a certain way, theoretically resistant since any assembly file can only be read inside the assigned device to it, meaning the original file can also only be accessed inside the user's device, and therefore, even if an attacker recurs to, and not exclusively, Sniffing, Eavesdropping or Man in the Middle (MiTM) techniques, he has to be able to get at the same time the user's wallet ID, a specific device serial number and it's MAC address in order to be able to access a specific user-owned file. If the file is not owned by that user, on that device, he will then also need to have this user's private key for the assembly file decryption, which has to be made inside the user's device and ran inside it as well, since the shared folder where that file is can not be moved out of the multi-dimensional matrix. All these steps are not impossible to achieve if there is enough motivation, time and resources available,

but nevertheless, even if they are indeed achieved, the fact that any file can not be opened or accessed outside the user's device makes it useless from an attacker side. Moreover, if in fact several mitigation mechanisms are also put in place as a compliment to this solution, then a nearly 100% brute force attack resistance would be achieved.

4.2.0.4 Phishing Attacks

Both the United Kingdom's (UK) Government Communications Headquarters (GCHQ) and the Computer Emergency Response Team (CERT) agree that Phishing attacks represent, nowadays, one of the biggest threats to information security and emphasize in one of their last joint works that roughly 80% of large companies had reported at least one security breach in 2014, representing each one of those breaches an estimated average cost between £600.000 and £1,6M [94]. CISCO, on the other hand, goes a bit further and states that phishing attacks are also one of the main precursors for both Advanced Persistent Threats (APTs) and Ransomware attacks while defining them as "the practice of sending fraudulent communications that appear to come from a reputable source" [95]. Their main purpose is, according to the USA's Office of the Director of National Intelligence (DNI), to acquire personal information or access to a computer system [96] and are, without a doubt, a very serious threat to consider in this work since they can be sent indiscriminately to several users within the network or, recurring to several methods and information gathering techniques like Social Engineering or even Open-Source Intelligence (OSINT) tools like Maltego or Shodan, they can be targeted to specific recipients after a deeper knowledge of the target is acquired. This more specific attack goes by the name of Spear-Phishing and according to the DNI, the typical *Modus Operandi* (MO) of these attacks consist first on an e-mail or message sent to a recipient with details that may get their attention, with a fraudulent link or attached file that is meant for them to open. The links will typically redirect to fraudulent pages in order to get users account's details for further fraudulent access, and the files will typically contain malware to get, in most of the cases, remote access to accounts or even machines, later leading to

potential integrity compromise of a computer, a network it resides on, and data [96] [97].

Taking into account the solution's architecture presented in this work it is possible that in some cases, a phishing attack and its variants, namely spear phishing, deceptive phishing, whaling and farming [95], if successful, would not represent a threat to a user's data. It is, of course, hard to mitigate these types of attacks if there is not a proper understanding of their MO and objectives, as well as a proper awareness on the user side, so if the victim of such an attack discloses all the necessary data to an attacker, all he will need is an emulator in his machine to mimic the user ID and the device ID in order to replicate the victim's credentials inside the application. This, however, would only be possible if malware is downloaded into a user's device by means of an attached file, since that is almost impossible for anyone to memorize 3 hashed values, which are not even disclosed on the display but are rather read directly inside the application after the calculations, plus the user private key; the only publicly known value from a user-side perspective. In other words, it is very unlikely and probably humanely impossible to type every correct hashed value by memory on a fraudulent link sent by e-mail or by any other means of communication, specially if not disclosed (which is the case), and in the event of downloaded malware, tools are already available to the public that would either stop or prevent the download or, in the event the malware is in fact downloaded, it would prevent it from running. These End-Point Detection and Response (EDR) or even SIEM tools, like Cybereason or Exabeam, when installed in a device, have the ability to prevent almost any attack due to its real-time response, aided by machine learning and deep learning techniques, meaning that the only possible attack vector, if a user is being aided by these tools, is a legit and conscious action from the user which would be humanely impossible to happen with the solution's architecture presented throughout chapter 3.

4.3 Other solution's security characteristics and resilience comparison

After the assessment made in section 4.1, the following section will focus on the assessment of other publicly available cloud storage solutions and will take into account every relevant security aspect so that a verifiable comparison can be made by means of a comparison matrix between the solution proposed in this work and others such as:

1. **AWS**
2. **Microsoft Azure**
3. **Storj**
4. **Tresorit**
5. **Boxcryptor**

4.3.0.1 AWS

Amazon Web Services, through its Simple Storage Service (S3), provides a cloud storage service to its clients throughout the globe, and even though the company doesn't provide too many specifications regarding its system's architecture, it is known that it is possible to either store data both encrypted and not encrypted and that the encryption/decryption keys may not be on the users side but rather in Amazon's one, meaning that users won't have full control of their data and that their keys are managed by a third party [98]. This option has to be selected by the user instead of being automatic, meaning that if he is not aware of this feature he may not have full control of what is his. On top of this, not only user's documents are stored in Amazon's services but also user's PII like name, age, e-mail and other contacts, credit card numbers and so on.

Nevertheless, Amazon S3 offers its users the possibility to log everything that its being done with their data on their account, which provides good transparency, traceability and auditability on its service, however, these logs can only be managed and analyzed recurring to other tools such as AWStats or Splunk, meaning that Amazon S3 alone won't be able to provide this feature to its users and therefore, possible integrations with other services won't be considered for this purpose since an assessment is being made for a single service as a whole and not with other tools integrations.

Regarding data or account access, the login procedures are mainstream and do not offer anything new, meaning that if user credentials are stolen or disclosed, data can be easily accessed by anyone without any constraints since there is not an authentication process that relates users and devices just like in this work's proposed solution. Also, and while not being a fully decentralized service but instead one single cloud (plus redundancy) managed by a single service, if there is an outage, users may not be able to access their data. In fact, Amazon S3's SLA states that they cover a 99,99% availability a month [99] , which reflects in roughly 14 minutes of unavailability every 24 hours.

Nonetheless, it is of utmost importance to mention that besides all these security procedures Amazon puts in place, BitDefender states that out of all S3 servers about 7% are publicly accessible and roughly 35% are not encrypted. Also it is worth to mention that in the last year alone several data breaches in Amazon S3 servers led to a combined 300+ Million user's data exposed, including PII, full credit reports and passwords in plain text [100].

Taking all this in consideration, it can't be assumed that this proposed solution does not offer, even though solemnly theoretical, security, traceability and auditability enhancements regarding Amazon's S3 service. Granting users full control of all of their keys and passwords while storing every record on a blockchain and therefore achieving full file access traceability and auditability on a system that does not store a single, complete and meaningful personal file as a whole while encrypting every fragment and enforcing various encryption phases throughout the

file sharing process, the enhancements regarding Amazon S3 are meaningful and shouldn't be disregarded.

4.3.0.2 Microsoft Azure

Being part of world giant Microsoft, Azure provides users, among other services, cloud storage provision services in almost the same way as Amazon, like described above, however, due to the USA Patriot Act, the United States Government can access user's data, even if stored outside the USA and concerning non-USA citizens [101].

Nevertheless, and just like Amazon, Microsoft claims that Azure provides 2 types of encryption: inactive encryption, which is fully managed by Azure's services and meaning that the encryption keys will never be on the user's side, and user-side encryption, granting user's full control of their keys. With these 2 solutions, data is first encrypted in the user's device and only afterwards it is stored in Azure's servers, remaining like this until it is accessed by the user or by someone who the user granted access [102].

Also just like Amazon's S3, this service lacks in decentralization and if a user wants file redundancy he has to choose from 4 different data replication categories with different features. Apart from this, everything remains the exact same in terms of offers regarding the above mentioned solution and other features like DDoS protection, although in this case it is not free like in Amazon S3 but instead it is payed and includes a proprietary Security Information and Event Management (SIEM) [103] .

Based on this, the same assumptions regarding data security, traceability and auditability will be made when in comparison with the previous solution and therefore the same conclusions regarding the theoretical solution presented in this work.

4.3.0.3 Storj

Storj is a decentralized cloud storage solution that relies on the Ethereum Blockchain and on its community members to store other people's data with end-to-end encryption features. Rewarding its users who provide both computational power and storage space, the so called "Farmers", with Storj Tokens as a payment, the platform also has the ability to be scaled up with other platforms and services like Amazon S3 through Application Programming Interfaces (APIs) and on top of this supports on-premise cloud storage solutions meaning it is suitable for public and hybrid storage solutions. Due to its implementation however, it is not possible to be ran solemnly on a (set of) private cloud(s). Taking this into account and comparing with the system, security and compliance requirements listed above in section 3.1, it is noticeable that, despite being built on a Ethereum blockchain, the files are stored as a whole, after being encrypted on a end-to-end basis, on other user's devices who have spare Hard Drive (HD) storage space and the legit owner of those files does not know who those users are unless if using the Amazon S3 API. In this case, the files are then stored in Amazon's services meaning that it is no longer a fully distributed and decentralised service.

Also, and admitting the files are stored in a fully decentralized way, i.e, in "farmers" computers, no one can be fully assured of the type of security these users have in place in their devices, meaning that if a computer is compromised, other user's files may also be. On the other hand, and despite working on a Zero-Knowledge basis, which offers pretty good guarantees for the users, this storage solution does not offer the possibility to share files with other people (unless the owner discloses his key), contrary to Tresorit.

4.3.0.4 Tresorit

Tresorit is a solution similar to Storj in terms of client-side encryption, while encrypting user's files with AES-256 before uploading them into the cloud, in this case, a single, private cloud; Microsoft Azure's. On top of this encryption layer,

Tresorit assures that files are also secured by Hash-based Message Authentication Codes (HMAC) applied on SHA-512, assuring its users that their data will keep and remain private and undisclosed. In fact, Tresorit's encryption method remains until today unbroken, which gives it's users very good guarantees of their service [104], however, it does not work on a fully distributed, decentralized way and relies on a single cloud service.

Based on the requirements listed in 3.1 in order to achieve the 3 above mentioned objectives, we can see that due to its encryption methods the offered security is very good, plus redundancy is guaranteed due to Microsoft Azure's Service Level Agreements (SLA), however in terms of traceability and auditability, and since no information about the document's and/or user's activity is stored anywhere, nothing can be traced except if that information is logged, and that is not disclosed by the 2 companies. Nevertheless, it can not be assumed that Tresorit's system is able to uniquely identify a user or a device because there is no mechanism to assure that. Every user can access their files in every device, which although is very good in terms of availability, it does not provide a better approach than the one presented in this work since everything can be uniquely identified, is recorded in the blockchain and is permanently immutable.

While not relying on a multi-cloud approach but instead on a single one, with the entire files stored as a whole, that may bring some issues if either the CSP is hacked and there is a data breach or if there is an unexpected downtime and the users are not able to access their data.

4.3.0.5 Boxcryptor

Just like Tresorit, Boxcryptor offers a very similar service with end-to-end encryption, meaning that only the legit file's owners, or someone they allow, will be able to access them and collaboratively work on them.

This is made possible because, just like Tresorit, the service relies on 3rd party CSPs in a fully centralized way. In this case, the user just needs to use Boxcryptor's software and it will automatically detect which CSP the client is using, making a "bridge" between the user's device and the storage service. Boxcryptor is used solemnly to encrypt the data and handle it to the user-chosen CSP, so every assumption made previously in 4.3.0.4 regarding security, traceability and auditability requirements can also be assumed the same way for boxcryptor.

4.3.1 Proposed solution's service availability

When talking about having our data timely available, on-demand, we are talking almost of a basic need people and businesses have nowadays and if businesses already depend both on structured and unstructured data made available, relying on sophisticated Information Technology (IT) infrastructures, for economic progress or to generate profit [105], people on the other hand use it for self satisfaction, extraversion, emotional stability and openness to experience [106]. Therefore, when a service intended to answer these expectations defrauds them, it comes with no surprise that people tend to discard it. Cloud computing brought a new paradigm to this matter, when it made data available anywhere, in any device, even if it was not physically in it, as long as there was an opened connection to the internet. This was made possible through service providers who provided the infrastructural means to do so, but nevertheless, the availability problem would remain if the service was centralized in a single service provider, as emphasized in [107]. To mitigate this issue, it is proposed to replicate data on multiple nodes, so that both response times and data availability would see a significant improvement [108], but some authors raise concerns about efficiency issues, when saying they might arise in peer-to-peer systems due to unreliable network conectivity, limited bandwith or erratic node failure [107].

Due to this solution's approach to peer-to-peer systems with blockchain technology and multi-cloud storage architecture, it is important to assess availability issues both in connection, authentication, and data access, so in order to verify

if the proposed solution does not offer full service availability, being the service comprised by all of the 3 components mentioned previously, 3 different aspects related to the solution's architecture will be scrutinized; the authentication process in the blockchain, the multi-cloud storage approach, regardless if its a public, a private or a hybrid architecture, and the shared folders and it's associated shared files redundancy.

When authenticating both users and devices in the blockchain, it is demonstrated in [109] that in several mainstream proof-of-work blockchains, reading availability is typically very high, while on the other hand, write availability is actually low , meaning that, and regarding the proposed solution, if a user and a device are already registered in the blockchain, i.e., there is a User ID, a Device ID and an ID Pair recorded in the blockchain, typically a user can login relatively quickly due to the blockchain's reading availability, nevertheles, and due to the blockchain's writting availability, when a user initiates it's registration and it's device's registration processes, that may take a while due to the block's generation and the time it takes to compute the nonce. Despite the time needed for both the reading and the recording processes for authentication purposes, it is of utmost importance to say that since we are talking of a highly decentralized infrastructure, regardless of its architecture, there will probably be no down-times whatsoever, assuring that even though both the registration and authentication processes may take more time than desired, these processes will happen since the infrastructure will typically never be shutdown, guaranteeing therefore the system's full availability.

As on what the solution's distributed, multi-cloud architecture concerns, and since it relies on several CSPs to deliver the service, with multiple redundancy copies distributed through every CSP in use, data would be fully available to its users as long as they had legit access to the intended assembly files and all the CSP in use were up and running. In every CSP in use, there would be a copy of the shared folder between the users and inside it, the assembly files shared between them, meaning that if 5 CSPs were in use and 2 users have shared between them

3 files, there would be at least 5 shared folders (at least 1 for CSP), with a copy of every assembly file inside it, i.e, 15 assembly files, 3 in each CSP.

Also, besides the shared folders and the assembly files inside them, in every CSP, the original file's fragments and its several redundancy copies are also distributed throughout all the CSPs in use, meaning that as long as the CSPs are not all down, users would be able to retrieve or access their data anytime they wanted as long as they have a valid access to the solution, both the user's and their associated device's are authenticated and there is a network connection.

Based on the above assumptions regarding the theoretical solution presented in this work, it can be said that even though the registration process might be slow and that the authentication process might take substantially less time than while registering for the first time, due to it's blockchain implementation, and therefore, being implemented in a highly distributed system, the overall availability of the system would be guaranteed as long as there is a continuous network connection.

As on what CSPs and stored data availability concerns, it can be stated that based on the above assumptions and on the redundancy approach mentioned in 3.2.4, if more redundancy CSPs are used and, therefore, more redundancy copies are generated on a 1 by 1 basis (one copy for each CSP), data availability would be virtually guaranteed unless all the CSPs were down at the same time, which is highly unlikely due to the service levels most of these services have. AWS, for instance, is designed to guarantee a 99,99% availability throughout a year, and just like Amazon, Google, Microsoft and others have the same standards, meaning that when combined between them we can assume, at least, the same guaranteed 99,99% availability and then disproving the initial hypothesis.

Theoretically, the proposed solution does provide the means to offer a full service availability.

4.3.2 Security characteristics considerations and comparison

Taking into consideration the assumptions made above, which are graphically depicted below in figure 4.1, regarding increased security, traceability and auditability offer by the present solution, it is plausible to conclude, although theoretically, that the null hypothesis can be disproved and therefore be assumed that the solution presented in this work does, indeed, provide the means to achieve an increased security, auditability and traceability when compared to other information security solutions available nowadays.

	Amazon S3	Microsoft Azure	Storj	Tresorit	BoxCryptor
Security	+++	+++	+	+	+
Traceability	+	+	NNI	+	+
Auditability	+	+	NNI	+	+

NNI	No Noticeable Improvements
+	Light Improvements
++	Substancial Improvements
+++	Very Substancial Improvements

FIGURE 4.1: Enhancements comparison

4.3.3 Resilience comparison

Being able to deal with unexpected problems, to quickly adapt to changes and overcome difficulties in a timely and efficient manner is subject of utmost importance in today's society, and while talking about cyber attack's resilience it comes with no surprise that, due to today's information systems inter connectivity and the way they impact our modern society, finding a way to resist the most complex forms of attacks is subject of the most relevant concerns and therefore, a study

of the possible positive impacts this solution may bring, when compared to other similar solutions, is considered in this section.

4.3.3.1 AWS

Having had several security and data breaches in the past, Amazon S3's service is, nevertheless, one of the most used worldwide and therefore the importance of assessing the theoretical resilience of S3 against different types of attacks.

On October 21st 2016 Amazon Web Services suffered a DDoS attack that led to the service's disruption for a considered amount of time and leaving user's without access to their data [110], but later in that year introduced Amazon Shield; a managed DDoS protection for AWS [111] which is free for S3 users. Due to this integration, Amazon increased DDoS attacks resilience considerably, nevertheless, the fact that user's data lies on a single server (plus redundancy) makes this solution more vulnerable than the one considered in this work, which is fully decentralized and without a central management entity.

Ransomware attacks won't be likely either and user's data will be easily recovered if such an attack occurs in their devices, but on the other hand, if a user has a weak or easily crackable password, brute force attacks might represent a serious problem, specially if MFA is not enabled. By gaining illegitimate access to an account, and offering Amazon the possibility for its users to store data in clear, without any type of encryption whatsoever, security issues may rise and a Brute Force attack might lead to persistent data breaches and unauthorised persistent account accesses while not preventing or even being able to mitigate this issue. It is, therefore, a major concern and just like a Brute Force attack, phishing and spear phishing attacks may also represent a serious problem having Amazon no possibility to prevent these types of attacks (which may very likely be, ultimately, the end-user's responsability).

Taking this into consideration, it is fair to say that theoretically, the security solution presented in this work offers the means to provide increased cyber attacks resilience when compared to Amazon S3's service.

4.3.3.2 Microsoft Azure

Based on the service's description in section 4.3.0.2, and taking into account the similarities of both Amazon S3 and Microsoft Azure solutions, the same assumptions as the ones made above in the previous section regarding cyber attack's resilience can also be made in the same way to Microsoft Azure's solution, although it has a considerable lack of DDoS protection when compared to S3.

This distinction shall be made because even though the 2 services offer proprietary DDoS mitigation and protection tools, Amazon Shield is free for S3 users while Azure's DDoS Protection is payed, meaning that even though these are 2 separate tools, on the first case it was considered due to its automatic integration while on this case it shouldn't be considered because only a few users, the ones who pay for the service, will have and therefore resilience could only be accounted for some users.

4.3.3.3 Storj

Based on the features previously mentioned in section 4.3.0.3, it is safe to say that Storj provides a very good resilience regarding DDoS attacks if kept on a fully distributed implementation. Only in a worst case scenario, it is possible that a user is not able to access one, or some, of his files if the device where they are stored is compromised in a certain way and/or the connection is somehow interrupted or denied due to a DDoS attack, since they are kept as a whole in the system. If, however, good redundancy is assured, this issue is then mitigated.

Regarding ransomware attacks, and if an Operating System (OS) is not securely patched for vulnerabilities, it may be the case that everything stored on a farmer's device gets corrupted and encrypted, loosing the owners access to these

particular files, possibly forever. Once again, if proper redundancy is not put in place, the owners of the files stored in every corrupted machine might loose their access forever without the possibility to properly recover them.

As for both brute force and phishing attacks, and due to its Zero-Knowledge encryption, even if an attacker is able to crack the farmer's password(s) and is able to retrieve, or copy, Storj's client's files form his device(s), he shouldn't be able to view or corrupt them since he doesn't have the file's decryption key. Nevertheless, it may be the case, even though not likely, that an attacker tries to break the file's decryption key so that he is then able to access it, but since that due to its encryption, which is unique for every user, no one knows who the file belongs to, it seems quite hard someone to spend so much resources in order to access a single, or maybe a few, files. Spear phishing attacks would then lose all its purpose, since an attacker would need to know who is who on the Ethereum blockchain, which goes against one of the blockchain's purpose (anonimity), and on top of that, he has to trace every victim's files to the respective Storj's farmers in order to try to get access to them and finally get access to those farmer's devices, retrieve the files and break its encryption. It is simply not likely at all, so it can be assumed that besides brute force and phishing attacks, spear phishing attacks would also not be a threat for this system, in the case the corrupted account is the farmer's. If, however, the corrupted account is the user's, and the attacker is able to access it knowing the correct key to decrypt data, then all the files within this user's account will be corrupted and, ultimately, access to them will be lost if the attacker deletes them, having no possibilities to have them back. Comparing Storj's Brute Force attacks and Phishing attack's resilience with the resilience provided by the solution presented in this work, one can see several improvements with this new system due to its authentication matrix which is able to mitigate these issues at a certain level. Instead of solemnly needing user's private keys, and contrary to Storj, this new system would not only require his private keys and account password, but every ID he has associated to him on the blockchain so that everything could be emulated and an illegitimate access could be simulated. Not 100% infallible, but more secure than Storj's system.

4.3.3.4 Tresorit

Tresorit's capabilities listed in 4.3.0.4 make this solution somewhat less resilient than Storj in terms of the 4 mentioned attacks despite providing better data security. The fact that user's data is solemnly stored in Microsoft's servers as a whole, even though fully encrypted, may make this solution more appealing to attackers than the previous one, since it does not rely on a fully distributed system but instead on a centralized one. Therefore, a DDoS attack to Microsoft Azure's servers might turn this data temporarily unavailable to it's users more easily than on a fully decentralized solution like Storj or the one presented in this work.

On the other hand, and regarding ransomware attacks, since Tresorit does not rely on their users and customers devices to store other entities data like Storj, and therefore mitigating the risk of having unreliable people and devices running their service, but on Azure's Cloud instead, a ransomware attack might be harder in this case due to the security implementations Microsoft has in place. Also, since data is not stored in the user's device but in the cloud instead, even if one of the user's devices is corrupted he will still be able to safely retrieve his data from the cloud on some other people's device as long as he logs in into his account.

Brute Force and Phishing attacks however, would have the same issue as Storj's system regarding end-users. Regarding the service provider however, this case wouldn't apply because the system is purely different and nothing is required to be manually input (log in into the user's machine, for instance) like in Storj, who relies its service on normal citizens with normal devices, and which the security concerns are not as big as in companies like Microsoft. Overall, and even though very similar regarding these 2 types of attacks, it can be reckoned that Tresorit offers a better alternative regarding brute force and phishing attack's resilience, nonetheless, slightly behind this work's solution who assures better counter-measures to mitigate these issues.

4.3.3.5 Boxcryptor

Just like Tresorit, Boxcryptor relies on the exact same conceptual design and working principles and therefore, the assumptions made for Tresorit regarding cyber attack's resilience should be assumed in the same way for BoxCryptor.

4.3.3.6 Resilience comparison considerations

Based on the 4 attack scenarios described in 4.2, on the expected behaviour of the presented solution, and on the assumptions made regarding cyber attack's resilience from 5 publicly available data storage solutions, an assessment was made taking in consideration all the comparisons made between these solution's behaviours and demonstrated capabilities with the one presented in the this work, resulting in the table depicted below in figure 4.2.

It can then be said that based on these attack scenario comparisons, the null hypothesis can then be disproved meaning that the solution presented in this work does indeed provide the theoretical means for an improved cyber attack resilience when compared to others currently available in the market.

	Amazon S3	Microsoft Azure	Storj	Tresorit	BoxCryptor
Attack Scenario 1	+	++	NNI	+	+
Attack Scenario 2	NNI	NNI	+	NNI	NNI
Attack Scenario 3	+++	+++	+	+	+
Attack Scenario 4	+++	+++	+	+	+

NNI	No Noticeable Improvements
+	Light Improvements
++	Substancial Improvements
+++	Very Substancial Improvements

FIGURE 4.2: Resilience Comparison

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Followed by the passion of information security and Blockchain and Distributed Ledgers Technology, acquired when i was studying at Tallinn University of Technology during my Erasmus Exchange program, and by trying to find ways to keep data safe from undisclosed eyes, the opportunity to address these 2 issues came with this idea, which was now translated into a conceptual design in this work, and therefore, it was set as the main purpose of it to present an enhanced information security solution for off-site storage. Due to BDLT characteristics, a theoretical approach to a blockchain based and fully decentralized and distributed implementation was considered both for user and device authentication and data storage and redundancy. This way, and recurring also to a multi-cloud approach, these 3 features could then be achieved.

Although several difficulties showed up, specially regarding the authentication process, the multi-dimensional matrix approach was the solution found to address these issues, and by means of this authentication method which combines User IDs and Device IDs in an unambiguous way, through the ID Pair, it was then theoretically possible not only to assure data's confidentiality and integrity, but also data availability due to its multi-cloud and data redundancy approaches,

assuring users that even if one CSP was temporarily down, they could still rely on other CSPs by means of this proposed solution and its multi-cloud approach.

Another solved issue was data transfer and portability between different users; not only due to the multi-dimensional hash matrix but also due to the shared folders between different users, it is then theoretically assured that files are only shared from and to specific, registered user-device pairs, allowing the solution to be fully transparent regarding shared data between users. Through this method, the exact same levels of confidentiality and integrity can be theoretically achieved even if data is shared with other users while recurring to this type of architecture, while guaranteeing, although theoretically, full transparency, auditability and traceability regarding file transfers.

Nevertheless, and despite the constraints, it was possible to positively answer the initial research question and to theoretically demonstrate in this work that a more secure, auditable and traceable solution, with high availability and with better cyber attack's resilience is indeed possible to achieve and, if fully functional and implemented on a fully distributed and decentralized way, it can bring both businesses and individuals more control of their own data, making it very hard for third parties to access it in an unauthorized way.

All in all it was a challenging work. Trying to improve and enhance third party's data security, integrity and availability through these new technological approaches was not an easy task, specially when there are already so many good approaches and solutions, but combining some of them in order to achieve this goal was unquestionably very rewarding and it certainly contributed to my increased knowledge in these fields of study.

5.2 Future Work

It is certainly a work in progress, and several aspects of this architecture need to be better detailed, addressed and researched. Validating the cryptography types

chosen for this architecture, regarding efficiency, speed and cost-effectiveness would be an interesting topic to address. Also the authentication method, the multi-dimensional hash matrix, would be interesting to study regarding its efficacy both in this solution and in others. The problems people face nowadays with electronic equipment theft, like cellphones and computers, could probably be mitigated if such a matrix was used to unequivocally address one device to one specific user. This would likely bring portability problems, in terms of sharing the devices itself, if there could only be one user in each machine, but nevertheless, it could be an interesting topic to approach while studying people's acceptance on a solution with this device portability limitation.

Blockchain and Distributed Ledgers could also be a study subject in the fields of information security, specially when addressed to cloud storage solutions on a larger scale, i.e, for companies. A fully decentralized cloud service would be the ultimate goal, but several topics need to be addressed such as blockchains processing speed and storage capacity. Assessing this systems feasibility with today's technology versus a system like the one proposed could also be subject of study within these fields in order to achieve the best possible solution and also, from another different perspective, a possible integration with Identity and Access Management tools in an enterprise environment should also be considered, possibly eliminating all the complex chain of events so that a user can access specific tools inside the company and therefore eliminating excessive work and time.

Ultimately, the main goal is to always guarantee that people have full control of their data and that they can access it regardless of where they are and when they want to do it, being at the same time fully assured it will keep and remain private.

Bibliography

[1] "UK businesses face growing threat from cyber-attacks – report." https://www.theguardian.com/technology/2018/apr/10/uk-businesses-face-growing-threat-from-cyber-attacks-report. Accessed: 2018-10-12.

[2] "2018 Internet Security Threat Report." https://www.symantec.com/security-center/threat-report. Accessed: 2018-10-12.

[3] "2017 Cybercrime Report." https://cybersecurityventures.com/2015-wp/wp-content/uploads/2017/10/2017-Cybercrime-Report.pdf. Accessed: 2018-10-12.

[4] B. Cashell, W. D. Jackson, M. Jickling, and B. Webel, "The economic impact of cyber-attacks," *Congressional Research Service Documents, CRS RL32331 (Washington DC)*, 2004.

[5] U. Bojars, A. Passant, and J. Breslin, "Data portability with sioc and foaf," 2008.

[6] A. Ranabahu and A. Sheth, "Semantics centric solutions for application and data portability in cloud computing," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pp. 234–241, IEEE, 2010.

[7] H. Chen, R. H. Chiang, and V. C. Storey, "Business intelligence and analytics: from big data to big impact," *MIS quarterly*, pp. 1165–1188, 2012.

[8] S. John Walker, "Big data: A revolution that will transform how we live, work, and think," 2014.

[9] A. Hevner and S. Chatterjee, "Design science research in information systems," in *Design research in information systems*, pp. 9–22, Springer, 2010.

[10] P. Mell, T. Grance, *et al.*, "The nist definition of cloud computing," 2011.

[11] H. Takabi, J. B. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24–31, 2010.

[12] M. Godse and S. Mulik, "An approach for selecting software-as-a-service (saas) product," in *Cloud Computing, 2009. CLOUD'09. IEEE International Conference on*, pp. 155–158, IEEE, 2009.

[13] J.-M. Bohli, N. Gruschka, M. Jensen, L. L. Iacono, and N. Marnau, "Security and privacy-enhancing multicloud architectures," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 4, pp. 212–224, 2013.

[14] M. A. AlZain, E. Pardede, B. Soh, and J. A. Thom, "Cloud computing security: from single to multi-clouds," in *System Science (HICSS), 2012 45th Hawaii International Conference on*, pp. 5490–5499, IEEE, 2012.

[15] S. Kamara, K. E. Lauter, *et al.*, "Cryptographic cloud storage.," in *Financial Cryptography Workshops*, vol. 6054, pp. 136–149, Springer, 2010.

[16] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation computer systems*, vol. 28, no. 3, pp. 583–592, 2012.

[17] "Big Data - 20 mind-boggling facts everyone must read." `https://www.forbes.com/sites/bernardmarr/2015/09/30/big-data-20-mind-boggling-facts-everyone-must-read/#4143536417b1`. Accessed: 2017-12-22.

[18] "Gartner - 8.4 billion "things" will be connected in 2017." `https://www.gartner.com/newsroom/id/3598917`. Accessed: 2017-12-22.

[19] "Big Data - business impacts." https://www.forbes.com/sites/bernardmarr/2015/09/08/4-ways-big-data-will-change-every-business/#2c9edc32729a. Accessed: 2017-12-22.

[20] "Big Data - life impacts." https://businessintelligence.com/bi-insights/7-ways-big-data-affects-everyday-life/. Accessed: 2017-12-22.

[21] "Big Data - not heard of life impacts." https://www.forbes.com/sites/peterpham/2015/08/28/the-impacts-of-big-data-that-you-may-not-have-heard-of/. Accessed: 2017-12-22.

[22] "Security and cost are barriers to adopting cloud tech." https://www.irishtimes.com/business/technology/security-and-cost-are-barriers-to-adopting-cloud-tech-1.3086169. Accessed: 2018-10-27.

[23] "Cloud computing in 2017: Adoption, Barriers, Trends." https://www.linkedin.com/pulse/cloud-computing-2017-adoption-barriers-trends-alexander-kupers/. Accessed: 2018-10-27.

[24] A. Celesti, M. Fazio, M. Villari, and A. Puliafito, "Adding long-term availability, obfuscation, and encryption to multi-cloud storage systems," *Journal of Network and Computer Applications*, vol. 59, pp. 208–218, 2016.

[25] T. G. Papaioannou, N. Bonvin, and K. Aberer, "Scalia: an adaptive scheme for efficient multi-cloud storage," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, p. 20, IEEE Computer Society Press, 2012.

[26] B. Alvi, M. Wasim Qureshi, and S. Karim, "Importance of information availability: its effects on business & the proposed model," *SSU Res. J. Eng. Technol*, vol. 1, no. 1, pp. 17–21, 2011.

[27] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems.," *NCM*, vol. 9, pp. 44–51, 2009.

[28] "2018 Security Report." `https://www.checkpoint.com/downloads/product-related/report/2018-security-report.pdf`. Accessed: 2018-10-12.

[29] "2017 Cost of Cyber Crime Study." `https://www.accenture.com/t20170926T072837Z__w__/us-en/_acnmedia/PDF-61/Accenture-2017-CostCyberCrimeStudy.pdf`. Accessed: 2018-10-12.

[30] K. D. Bowers, A. Juels, and A. Oprea, "Hail: A high-availability and integrity layer for cloud storage," in *Proceedings of the 16th ACM conference on Computer and communications security*, pp. 187–198, ACM, 2009.

[31] V. Balasaraswathi and S. Manikandan, "Enhanced security for multi-cloud storage using cryptographic data splitting with dynamic approach," in *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on*, pp. 1190–1194, IEEE, 2014.

[32] M. A. AlZain, B. Soh, and E. Pardede, "Mcdb: using multi-clouds to ensure security in cloud computing," in *Dependable, autonomic and secure computing (DASC), 2011 IEEE Ninth International Conference on*, pp. 784–791, IEEE, 2011.

[33] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "Depsky: dependable and secure storage in a cloud-of-clouds," *ACM Transactions on Storage (TOS)*, vol. 9, no. 4, p. 12, 2013.

[34] "NSA - security breach and spilled secrets have shaken the n.s.a. to its core." `hhttps://www.nytimes.com/2017/11/12/us/nsa-shadow-brokers.html`. Accessed: 2017-12-18.

[35] "iCloud data breach: Hacking and celebrity photos." `https://www.forbes.com/sites/davelewis/2014/09/02/`

icloud-data-breach-hacking-and-nude-celebrity-photos/#3604f8a72de7. Accessed: 2017-12-18.

[36] "Every single Yahoo account was hacked - 3 billion in all." https://money.cnn.com/2017/10/03/technology/business/yahoo-breach-3-billion-accounts/index.html. Accessed: 2018-10-27.

[37] B. Von Solms and R. Von Solms, "The 10 deadly sins of information security management," *Computers & Security*, vol. 23, no. 5, pp. 371–376, 2004.

[38] C. J. Alberts and A. Dorofee, *Managing information security risks: the OCTAVE approach*. Addison-Wesley Longman Publishing Co., Inc., 2002.

[39] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of network and computer applications*, vol. 34, no. 1, pp. 1–11, 2011.

[40] "Threat, vulnerability, risk – commonly mixed up terms." https://www.threatanalysis.com/2010/05/03/threat-vulnerability-risk-commonly-mixed-up-terms/. Accessed: 2018-01-05.

[41] M. E. Whitman, "Enemy at the gate: threats to information security," *Communications of the ACM*, vol. 46, no. 8, pp. 91–95, 2003.

[42] X. Zhang, N. Wuwong, H. Li, and X. Zhang, "Information security risk management framework for the cloud computing environments," in *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, pp. 1328–1334, IEEE, 2010.

[43] "IBM Knowledge Center – benefits of high availability." https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzarj/rzarjbenefitsha.htm. Accessed: 2018-01-05.

[44] J. E. Boritz, "Is practitioners' views on core concepts of information integrity," *International Journal of Accounting Information Systems*, vol. 6, no. 4, pp. 260–279, 2005.

[45] E. McCallister, T. Grance, and K. A. Scarfone, "Guide to protecting the confidentiality of personally identifiable information (pii)," *Special Publication (NIST SP)-800-122*, 2010.

[46] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 1996.

[47] J. Ding and A. Petzoldt, "Current state of multivariate cryptography," *IEEE Security & Privacy*, vol. 15, no. 4, pp. 28–36, 2017.

[48] V. Kapoor, V. S. Abraham, and R. Singh, "Elliptic curve cryptography," *Ubiquity*, vol. 2008, no. May, p. 7, 2008.

[49] L. Chen and S. Zhou, "The comparisons between public key and symmetric key cryptography in protecting storage systems," in *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, vol. 4, pp. V4–494, IEEE, 2010.

[50] G. Zyskind, O. Nathan, *et al.*, "Decentralizing privacy: Using blockchain to protect personal data," in *Security and Privacy Workshops (SPW), 2015 IEEE*, pp. 180–184, IEEE, 2015.

[51] "Harvard Business Review – the truth about blockchain." `https://hbr.org/2017/01/the-truth-about-blockchain`. Accessed: 2018-01-06.

[52] "Bitcoin: a peer-to-peer electronic cash system." `https://bitcoin.org/bitcoin.pdf`. Accessed: 2018-01-06.

[53] "Ethereum Project." `https://www.ethereum.org`. Accessed: 2018-01-07.

[54] "Blockchain Asset Management: litecoin." `https://www.falconpb.com/documents/48599/111360/Litecoin+Introsheet.pdf/38951fcf-f637-e5ec-a400-cdfbc9a0af8c`. Accessed: 2018-01-07.

[55] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3–16, ACM, 2016.

[56] A. Kiayias and G. Panagiotakos, "Speed-security tradeoffs in blockchain protocols.," *IACR Cryptology ePrint Archive*, vol. 2015, p. 1019, 2015.

[57] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, 2017.

[58] A. Anjum, M. Sporny, and A. Sill, "Blockchain standards for compliance and trust," *IEEE Cloud Computing*, vol. 4, no. 4, pp. 84–90, 2017.

[59] N. Kshetri, "Blockchain's roles in strengthening cybersecurity and protecting privacy," *Telecommunications Policy*, 2017.

[60] M. Pilkington, "Blockchain technology: principles and applications," *Browser Download This Paper*, 2015.

[61] "Blockchain Technology and Cryptocurrencies: The Financial Revolution And The Many Benefits It Brings." `https://www.forbes.com/sites/forbescommunicationscouncil/2017/11/15/the-financial-revolution-and-the-many-benefits-it-brings-cryptocurrency-bl#4bafb4ae3cc0`. Accessed: 2018-01-08.

[62] "How blockchains could change the world." `https://www.mckinsey.com/industries/high-tech/our-insights/how-blockchains-could-change-the-world`. Accessed: 2018-01-08.

[63] M. E. Peck, "Blockchain world-do you need a blockchain? this chart will tell you if the technology can solve your problem," *IEEE Spectrum*, vol. 54, no. 10, pp. 38–60, 2017.

[64] "Estonia wants to launch its own government-backed cryptocurrency." `https://www.cnbc.com/2017/08/23/estonia-cryptocurrency-called-estcoin.html`. Accessed: 2018-01-08.

[65] "Crypto Market Value Surpasses 500 Billion Dollar As Interest Surges." `https://www.forbes.com/sites/cbovaird/2017/12/12/crypto-market-value-surpasses-500-billion-as-interest-surges/#fdc8e033067a`. Accessed: 2018-01-08.

[66] D. Shrier, W. Wu, and A. Pentland, "Blockchain & infrastructure (identity, data security)," tech. rep., Retrieved 27-11-16, from http://cdn. resources. getsmarter. ac/wp-content/uploads/2016/06/MIT_Blockain_Whitepaper_PartThree. pdf, 2016.

[67] "STORJ - Decentralized Cloud Storage." `https://storj.io`. Accessed: 2018-01-09.

[68] S. Wilkinson, J. Lowry, and T. Boshevski, "Metadisk a blockchain-based decentralized file storage application," tech. rep., Technical Report, Available: http://metadisk. org/metadisk. pdf, 2014.

[69] "Boxcryptor | Security for your Cloud." `https://www.boxcryptor.com/en/`. Accessed: 2018-10-27.

[70] "Tresorit: End-to-End Encrypted File Sync and Sharing." `https://tresorit.com`. Accessed: 2018-10-27.

[71] "How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read." `https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-rea#2ee955f860ba`. Accessed: 2018-10-12.

[72] F. Mendel, T. Nad, and M. Schläffer, "Improving local collisions: new attacks on reduced sha-256," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 262–278, Springer, 2013.

[73] S. K. Sanadhya and P. Sarkar, "New collision attacks against up to 24-step sha-2," in *International conference on cryptology in India*, pp. 91–103, Springer, 2008.

[74] K. Aoki, J. Guo, K. Matusiewicz, Y. Sasaki, and L. Wang, "Preimages for step-reduced sha-2," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 578–597, Springer, 2009.

[75] D. G. Amalarethinam and J. S. Geetha, "Image encryption and decryption in public key cryptography based on mr," in *Computing and Communications Technologies (ICCCT), 2015 International Conference on*, pp. 133–138, IEEE, 2015.

[76] W. W. Chu, "Optimal file allocation in a multiple computer system," *IEEE Transactions on Computers*, vol. 100, no. 10, pp. 885–889, 1969.

[77] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson, "Raid: High-performance, reliable secondary storage," *ACM Computing Surveys (CSUR)*, vol. 26, no. 2, pp. 145–185, 1994.

[78] "Advanced Threat Analytics." `https://www.microsoft.com/en-us/cloud-platform/advanced-threat-analytics`. Accessed: 2018-10-20.

[79] "Cybercrime will cost businesses over $2 trillion by 2019." `hhttps://www.juniperresearch.com/press/press-releases/cybercrime-cost-businesses-over-2trillion`. Accessed: 2018-10-20.

[80] "2018 Internet Security Threat Report." `https://www.symantec.com/security-center/threat-report`. Accessed: 2018-10-20.

[81] "Buffett: This is 'the number one problem with mankind'." `https://www.businessinsider.com/warren-buffett-cybersecurity-berkshire-hathaway-meeting-2017-5`. Accessed: 2018-10-20.

[82] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, "Controlling data in the cloud: outsourcing computation without outsourcing control," in *Proceedings of the 2009 ACM workshop on Cloud computing security*, pp. 85–90, ACM, 2009.

[83] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

[84] D. Larson, "Distributed denial of service attacks–holding back the flood," *Network Security*, vol. 2016, no. 3, pp. 5–7, 2016.

[85] S. Mohurle and M. Patil, "A brief study of wannacry threat: Ransomware attack 2017," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, 2017.

[86] X. Luo and Q. Liao, "Awareness education as the key to ransomware prevention," *Information Systems Security*, vol. 16, no. 4, pp. 195–202, 2007.

[87] R. Brewer, "Ransomware attacks: detection, prevention and cure," *Network Security*, vol. 2016, no. 9, pp. 5–9, 2016.

[88] "Ransomware Prevention and Response for CISOs." `https://www.fbi.gov/file-repository/ransomware-prevention-and-response-for-cisos.pdf/view`. Accessed: 2018-10-20.

[89] "What's a Brute Force Attack?." `https://www.kaspersky.com/resource-center/definitions/brute-force-attack`. Accessed: 2018-10-17.

[90] "Dictionary Attack." `https://www.techopedia.com/definition/1774/dictionary-attack`. Accessed: 2018-10-19.

[91] J. Owens and J. Matthews, "A study of passwords and methods used in brute-force ssh attacks," in *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2008.

[92] P. Oechslin, "Making a faster cryptanalytic time-memory trade-off," in *Annual International Cryptology Conference*, pp. 617–630, Springer, 2003.

[93] A. Narayanan and V. Shmatikov, "Fast dictionary attacks on passwords using time-space tradeoff," in *Proceedings of the 12th ACM conference on Computer and communications security*, pp. 364–372, ACM, 2005.

[94] "Common Cyber Attacks: Reducing The Impact." `https://assets.publishing.service.gov.uk/government/uploads/system/uploads/`

attachment_data/file/400106/Common_Cyber_Attacks-Reducing_The_Impact.pdf. Accessed: 2018-10-19.

[95] "What Is Phishing?." https://www.cisco.com/c/en/us/products/security/email-security/what-is-phishing.html. Accessed: 2018-10-19.

[96] "Spear Phishing and common Cyber Attacks." https://www.dni.gov/files/NCSC/documents/campaign/Counterintelligence_Tips_Spearphishing.pdf. Accessed: 2018-10-19.

[97] J. Hong, "The state of phishing attacks," *Communications of the ACM*, vol. 55, no. 1, pp. 74–81, 2012.

[98] "New Amazon S3 Encryption and Security Features." https://aws.amazon.com/pt/blogs/aws/new-amazon-s3-encryption-security-features/. Accessed: 2018-10-29.

[99] "Contrato de Nível de Serviço do Amazon S3." https://aws.amazon.com/pt/s3/sla/. Accessed: 2018-10-29.

[100] "10 Worst Amazon S3 Breaches." https://businessinsights.bitdefender.com/worst-amazon-breaches. Accessed: 2018-10-29.

[101] "Microsoft: European cloud data may not be immune to the Patriot Act." https://www.engadget.com/2011/06/30/microsoft-european-cloud-data-may-not-be-immune-to-the-patriot/?guccounter=1. Accessed: 2018-10-29.

[102] "Introdução ao Armazenamento do Azure." https://docs.microsoft.com/pt-pt/azure/storage/common/storage-introduction. Accessed: 2018-10-29.

[103] "Proteção contra DDoS do Azure." https://azure.microsoft.com/pt-pt/services/ddos-protection/. Accessed: 2018-10-29.

[104] "Tresorit: Protect your data with zero-knowledge encryption." https://tresorit.com/business/wuala-alternative. Accessed: 2018-10-28.

[105] F. X. Olleros and M. Zhegu, *Research handbook on digital transformations*. Edward Elgar Publishing, 2016.

[106] T. Correa, A. W. Hinsley, and H. G. De Zuniga, "Who interacts on the web?: The intersection of users' personality and social media use," *Computers in Human Behavior*, vol. 26, no. 2, pp. 247–253, 2010.

[107] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[108] K. Ranganathan, A. Iamnitchi, and I. Foster, "Improving data availability through dynamic model-driven replication in large peer-to-peer communities," in *Cluster Computing and the Grid, 2002. 2nd IEEE/ACM International Symposium on*, pp. 376–376, IEEE, 2002.

[109] I. Weber, V. Gramoli, A. Ponomarev, M. Staples, R. Holz, A. B. Tran, and P. Rimba, "On availability for blockchain-based systems," in *Reliable Distributed Systems (SRDS), 2017 IEEE 36th Symposium on*, pp. 64–73, IEEE, 2017.

[110] "Major DDoS attack on Dyn disrupts AWS, Twitter, Spotify and more ." `https://www.datacenterdynamics.com/news/major-ddos-attack-on-dyn-disrupts-aws-twitter-spotify-and-more/`. Accessed: 2018-10-29.

[111] "Introducing AWS Shield." `https://aws.amazon.com/pt/about-aws/whats-new/2016/12/introducing-aws-shield/`. Accessed: 2018-10-29.