future internet

MDPI

Review

Security and Privacy in Cloud Computing: Technical Review

Yunusa Simpa Abdulsalam * and Mustapha Hedabou

DNA Lab, School of Computer and Communication Science, University Mohammed VI Polytechnic, Lot 660, Hay Moulay Rachid, Ben Guerir 43150, Morocco; mustapha.hedabou@um6p.ma
* Correspondence: abdulsalam.yunusa@um6p.ma

Abstract: Advances in the usage of information and communication technologies (ICT) has given rise to the popularity and success of cloud computing. Cloud computing offers advantages and opportunities for business users to migrate and leverage the scalability of the pay-as-you-go price model. However, outsourcing information and business applications to the cloud or a third party raises security and privacy concerns, which have become critical in adopting cloud implementation and services. Researchers and affected organisations have proposed different security approaches in the literature to tackle the present security flaws. The literature also provides an extensive review of security and privacy issues in cloud computing. Unfortunately, the works provided in the literature lack the flexibility in mitigating multiple threats without conflicting with cloud security objectives. The literature has further focused on only highlighting security and privacy issues without providing adequate technical approaches to mitigate such security and privacy threats. Conversely, studies that offer technical solutions to security threats have failed to explain how such security threats exist. This paper aims to introduce security and privacy issues that demand an adaptive solution approach without conflicting with existing or future cloud security. This paper reviews different works in the literature, taking into account its adaptiveness in mitigating against future reoccurring threats and showing how cloud security conflicts have invalidated their proposed models. The article further presents the security threats surrounding cloud computing from a user perspective using the STRIDE approach. Additionally, it provides an analysis of different inefficient solutions in the literature and offers recommendations in terms of implementing a secure, adaptive cloud environment.

Keywords: cloud computing; security; privacy; privacy preserving

Citation: Abdulsalam, Y.S.A.; Hedabou, M. Security and Privacy in Cloud Computing: Technical Review. *Future Internet* **2022**, *14*, 11. https://doi.org/10.3390/fi14010011

Academic Editor: Massimo Cafaro

Received: 24 October 2021
Accepted: 7 December 2021
Published: 27 December 2021

1. Introduction

The Internet service industry, including areas such as cloud computing, is an evolving paradigm for large scale infrastructure [1]. Cloud computing possesses the power to reduce costs by resource sharing and storage virtualisation, collectively merged with a provisioning mechanism that relies on a pay-as-you-go business architecture [2]. Cloud computing technologies such as Amazon's Elastic Computing Cloud (EC2), Simple Storage Service (S3) and Google App Engine have been the most popular in the software industry. Despite the impact and the efficient services these applications have offered, there are still security and privacy issues relating to how these cloud providers process users' data [3]. Issues arising because of insecure cloud computing platforms spread across different technological paradigms such as web-based outsourcing [4], mobile cloud computing [5] and service-oriented architectures (SOA). Secure cloud implementation demands an adaptive security mechanism to help users have a significant level of trust in the cloud. Without the ability of such techniques to guarantee a substantial level of security and privacy, there will continue to be a great fear of privacy loss and sensitive data leakage, which are significant obstacles and a deciding factors in the full adoption of cloud services [1].

Privacy is a fundamental human right that comprises the right to be left alone and demands the appropriate use and protection of personal information [6]. The implementation of cloud computing paradigms violates privacy in different ways, such as misappropriation

of confidential information [7], uncontrollable use of cloud services, data propagation, potential unauthorised secondary usage, trans-border flow of data and dynamic provisioning. Other privacy concerns are data retention regulation, outsourced data deletion, and privacy awareness breaches [8]. In current practices, a consensus is typically achieved through a third-party service or by the general terms and conditions for personal data processing. The security and privacy issues become more complicated when granting user permission in an environment with minimal or no user interface due to unauthorised data usage permission and ineffective processing of personal information, which is often not considered during the designing phase. In terms of cloud security implementation, there are questions about data security policies for users in the cloud environment. Firstly, what are the commitments of Cloud Service Providers (CSPs) in establishing information security? Secondly, what data security policies have been published and made open to the public? The lack of clear justification has led to recent violations of privacy. In April 2019, Facebook Inc. was sued for a total of USD 5bn for Analytica privacy violations, making infrastructures for data security be under constant scrutiny to meet user privacy needs. Still, there has not been any clear direction for management support initiatives [9]. The authorisation process and access control mechanisms for data processing facilities have not been very efficient due to insider attacks generated from internal personnel. Most recently, organisations have been entrusting the security of users' confidential data to third-party access for security auditing, raising more security concerns on accountability of third-party. The best-case scenario is an honest but curious third party, which is still not suitable for real-life deployment [10]. Thirdly, what measures are defined to classify data access, and how can they be justified through third-party auditing? In granting third-party access, organisations need to define a hierarchy for accessing data, and proper identity management for third-party access should be an essential task for every CSP [9]. Without appropriate identity management, an inside attack can occur by deploying malicious applications on edge nodes, exploiting vulnerabilities that affect the quality of service (QoS). Such hostile acts can significantly affect sensitive data temporarily saved on multiple edge routers.

As more organisations are moving to the cloud as an effective means of data storage, they need to share, process rapidly and disseminate a high volume of sensitive information to enhance effective decision-making [11]. However, a significant setback is the lack of security and privacy flexibility. Current security and privacy mechanism lacks the flexibility in responding to the changing external environment, which has led to an uncontrollable risk of data leakage. Organisations are concerned about stabilising cloud security infrastructures without depleting data leakage and information of users. Unfortunately, data storage services keep changing and, today, privacy can be individually defined—what might be private for an individual might be disclosed by some without concern. Therefore, there is a need to describe non-specific requirements when building privacy and security protocols for cloud computing. Strict privacy or security protocols will only be stagnant in the long run because technology and its resources are moving to the open world where everyone might decide what they choose to be private, especially in the cloud environment.

This review aims to provide a technical approach for researchers who want to dive into the field of security and privacy for cloud computing, serving as a point of reference. Different reviews on cloud computing already exist in the literature. However, all have failed to provide a single report that brings a balance between security, privacy and a technical approach that provides a scientific insight into the different research gaps in cloud computing. Our specific contribution is as follows:

1. Understanding of the cloud computing concept in relation to user privacy and security.
2. Classification of cloud components, threats, and security implementations based on the STRIDE model.
3. Providing security and privacy classifications based on attack mitigation and adaptiveness.
4. Providing different approaches to what and how existing works in the literature have provided solutions to cloud computing security and privacy.

2. Background

Cloud security is a branch of computer and network security controlled by privacy-enhancing technologies and governed by a set of policy rules to protect the deployment of data, software applications, and associated services outsourced in the cloud [12]. Common terminologies in the field of security are shown in Table 1. These terminologies are used across all fields when defining the security and privacy of a particular research area. The STRIDE model [13] provides a systematic way of analysing vulnerabilities by providing distinct understanding based on technical knowledge [14]. The STRIDE approach of analysing vulnerabilities is a matching concept to the existing security terminologies, as shown in Table 2. The STRIDE model is an effective way of knowing the impact vector of an attack before its occurrence [15]. This approach has previously been used in the literature for accessing threat capacity in cloud computing. Literature review of cloud computing security and privacy is shown Table 3 and Table 4 respectively.

Table 1. Security parameter definition.

Terminology	Definition
Confidentiality	To ensure the accessibility of information to only authorised users.
Integrity	Maintaining the completeness and accuracy of every part of information.
Availability	Information is accessible to only authorised users.
Non-repudiation	Avoid the deniability of one's actions.
Privacy-preserving	Ability to mask identity and Personal Identifiable Information (PII).
Accountability	Obligation or willingness to take responsibility for action with a defined set of rules.
Auditability	Maintaining a system with relative ease in other to improve its efficiency.
Authentication	Establishing the right identity of a user in a system
Authorisation	Access to resources is restricted to only authorised personnel

Table 2. STRIDE security definition.

STRIDE Threat	Matching Security Parameter
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-repudiation
Information disclosure	Confidentiality
Denial of service	Availability
Elevation of privilege	Authorisation

Table 3. Review of literature I.

Reference	Reviewed Layer	Security	Privacy	Technical Approach	Remark
[16]	IaaS, PaaS, SaaS	✓	✓	×	Aimed at distinguishing the different aspects of cloud computing in order to better understand and present its security and privacy issues.
[17]	IaaS, PaaS, SaaS	✓	✓	×	Surveyed the different security factors affecting the adoption of cloud computing. Identified and provided solution perspectives to further strengthen its privacy and security.
[18]	IaaS	✓	×	✓	Threat in hardware and operating system virtualisation related to cloud computing. Accomplished by properly categorising trust assumptions, security and threat models.
[19]	IaaS, PaaS, SaaS	✓	×	×	Provided a comparison of other survey articles on the basis of computational, communication and service layer agreement level of cloud Cloud security challenges.
[20]	IaaS, PaaS, SaaS	✓	×	×	Provided the security issues in different service delivery layers that pose a threat to the adoption of cloud computing.
[21]	IaaS	✓	×	✓	Provided a state-of-the-art survey on approaches and solutions of current security trends on resource scheduling in cloud computing.
[22]	IaaS, PaaS, SaaS	✓	×	✓	Highlighted the necessary loop holes, security and privacy recommendations surrounding cloud computing. Presenting a generalised opinion on security and privacy flaws.
[23]	IaaS, PaaS, SaaS	×	✓	✓	Presented state-of-the-art introduction to cryptographic approach for privacy preserving in cloud computing, putting into perspective the adoption of online applications.
[24]	IaaS, PaaS, SaaS	✓	×	×	Provided insights on the future of cloud computing by highlighting technical and adoption issues that will present themselves without adequate security and privacy measures.
[25]	IaaS, PaaS, SaaS	✓	×	✓	Surveyed the privacy, security and trust issues surrounding cloud computing and further provided possible cryptographic solutions.
[26]	SaaS	✓	✓	✓	Analysis on key management and secure practices on cryptographic operations in the cloud.

Table 4. Review of literature II.

Reference	Reviewed Layer	Security	Privacy	Technical Approach	Remark
[27]	PaaS, SaaS	✓	✓	✓	Reviewed data storage integrity and auditing in cloud computing by highlighting state-of-the-art methods and challenges.
[28]	IaaS, PaaS, SaaS	✓	×	✓	Discussed and presented state-of-the-art task scheduling security issues and limitations in cloud computing, based on application, methods and utilisation.
[29]	PaaS, SaaS	✓	✓	×	Presented the threats and vulnerabilities open to attackers in cloud computing by considering accountability, integrity, availability, confidentiality and privacy preserving.
[30]	PaaS, SaaS	✓	×	✓	Presented an extensive review on outsourced data bases in cloud computing introducing new database query and encryption.
[31]	PaaS, SaaS	✓	✓	✓	Classified state-of-the-art taxonomy on current remote data auditing scheme and their limitations based on security metrics and requirements, data update and auditing.
[32]	IaaS, PaaS, SaaS	✓	✓	×	Presented issues of trust, security and privacy in cloud computing by assessing the different factors that affect its adoption.
[33]	PaaS, SaaS	✓	×	✓	Surveyed remote data integrity and auditing in cloud computing. Providing an enhancement to the review literature of [34]
[35]	IaaS, PaaS, SaaS	✓	✓	×	Presented trends and research directions in cloud computing by considering computing models that are prone to threats and vulnerabilities.
[36]	IaaS, PaaS, SaaS	✓	✓	×	Analysed privacy and security issues in cloud computing by considering the different components and relationship to organisational internet of things protocol.
[37]	IaaS, PaaS, SaaS	✓	✓	×	Provided a taxonomy of security and privacy and further presented several attack detection remedies in cloud computing.
[34]	IaaS, PaaS, SaaS	✓	✓	×	Provided a taxonomy on remote data auditing and integrity in cloud computing by analysing data replication, erasure and communication.

2.1. Cloud Computing Service Delivery Models

The flexibility of cloud infrastructure and economic benefits has become the great motivations in continuing to adopt the cloud [38]. Additionally, cloud infrastructure has provided computing power and resource scalability [39]. As a result, it has provided ubiquitous network access, independent resource pooling, on-demand self-service, usage-based pricing, and resource elasticity. Definition 1 provides a technical description of a Cloud Service *cs* from a CSP [40].

Definition 1. *Cloud Service: Let cs denote a cloud service hosted in the cloud with an identity of* CI_{id}*, being consumed by users through a set of interface i, defined by some APIs in the Internet: Then* $cs = [i, CP, CI_{id}]$*, where CP is a set of collaboration processes that cs relies on during service delivery.* CI_{id} *is denoted as the ID of the underlying cloud infrastructure. Through this definition, we can define the Amazon EC2 service as:* $ecs = [ec2, acp, wsdl, aws]$*, where* $acp = [create_{VM}, start_{VM}, connect_{VM}, stop_{VM}, cancel_{VM}]$*, and wsdl is the Web Service Description Language.*

The cloud computing infrastructure comprises three service delivery models that help promote the availability and virtualisation of resources [12]. The STRIDE analysis of the different service delivery models is shown in Table 5 and defined as follows:

1. Cloud Infrastructure as a Service (IaaS): IaaS provides aggregated resources managed physically. Service delivery is in the form of storage or computational capability. The IaaS platform offers storage, provision processing and networks for consumers to run and deploy arbitrary software for applications and operating systems. The platform user might not have absolute control over the underlying infrastructure but control the deployed applications, operating system, and network components. The IaaS layer represents the pillar for which most cloud computing architectures have been built [41]. As a result of high advancement in technology, computational power, storage devices and high-end communication, the IaaS layer has become the most efficient platform on which the PaaS and SaaS rely.
2. Cloud Platform as a Service (PaaS): PaaS provides platforms and programming environments for cloud infrastructure services. Examples of PaaS includes Google App Engine, Dipper, Yahoo and Salesforce. PaaS also refers to the application developed by a programming language and hosted by a CSP in the cloud [41]. PaaS is the service abstraction of the cloud that deals with the creation and modification of applications that already exist. The advantage of PaaS is provisioning platform environments with full operational and developmental features for application deployment. Furthermore, PaaS provides a trusted environment for users' secure storage and processing of confidential information, leveraged by the cryptographic co-processors [42] that protect against unauthorised access. The central design and goal of the PaaS are maximising user control when managing features related to the privacy of sensitive information, accomplished through user data privacy methods and self-installed configurable software.
3. Cloud Software as a Service (SaaS): SaaS provides confinement for client flexibility by providing software applications and APIs for developers such as GoogleMaps and Bloomberg. SaaS consumers are obliged to pay for software on a subscription basis, with no need for prior installations. Accessing SaaS software is primarily through the internet via a web browser. SaaS provides live applications running in the cloud, accessed through users' devices connected to the internet. Unlike the IaaS, SaaS user does not have control over storage, operating systems, network components, or the underlying infrastructure [41]. Its primary advantage is its multi-tenancy nature because it can share access control to the software.

Table 5. Cloud service delivery STRIDE analysis.

	Infrastructure as a Service	Platform as a Service	Software as a Service
Spoofing		X	X
Tampering			X
Repudiation			X
Information Disclosure			X
Denial of Service	X	X	X
Elevation of Privilege	X	X	X

The symbol X denotes the existence of a STRIDE component.

2.2. Cloud Computing Deployment Models

Organisations can deploy cloud computing infrastructure using four different architectures. Deployment depends on the ownership, administration, location, security policies, and nature of the data. The STRIDE analysis of the four cloud computing deployment models is shown in Table 6 and are as follows:

1. Private cloud: Deployment environment is owned by private sectors solely for the secure storage of company's data [41]. Private clouds are managed mainly by third-party providers but exist on-premise. Access is granted only by company staff to control authorisation management for security purposes. For example, an organisation that wants to make its customer's data available can create a private data centre. Providing more access control over sensitive information and enhanced data security mechanisms to ensure privacy in a private cloud setting. The major drawback of these settings is their purchase cost for equipment and utility bills.
2. Community cloud: A cloud environment collectively owned by a set of organisations with the same motive. The community cloud is similar to a private cloud, but the computational resources and underlying infrastructure are exclusively controlled by two organisations with common privacy and security motives. It is also more expensive than the public cloud, and data access is not regulated correctly due to untrusted parties that might arise. The advantage of the community cloud is the involvement of fair third-party access for security auditing.
3. Public cloud: The public cloud is mainly owned by large organisations offering cloud services, such as Google Apps, Amazon AWS and Microsoft Office 365. Resources in public clouds are primarily provided as a service at a pass-as-you-go fee. The benefits are mainly on-demand purchases: the more the usage, the more the payment. Public cloud users are mostly home users in their houses accessing the providers' network via the internet. The security issues of the public cloud are its lack of data security and privacy as a result of its public nature. There is no control over the transmission of information or the access to sensitive data [41]. Despite its colossal security limitation, small organisations have benefited from its services due to their limited sensitive information with minimal privacy risks.
4. Hybrid cloud: A hybrid cloud service can be offered by a private cloud owner forming a partnership with a public owner, making it more complex because of the involvement of two or more cloud providers. This approach allows the cost-effectiveness and scalability of public cloud environments without exposing data to third-party and mission-critical software applications. The hybrid system offers private cloud features, enabling rapid scalability features of the public cloud. Overall, it provides a drastic improvement to organisational agility and offers greater flexibility to business when compared to other approaches. The security limitations of the hybrid cloud are the limitations of the public cloud, such as public exposure of sensitive information, which poses a significant security risk. An approach to solving this issue is the idea of identity and access management to cloud facilities.

Table 6. Cloud deployment STRIDE analysis.

	Private Cloud	Community Cloud	Public Cloud	Hybrid Cloud
Spoofing		X	X	X
Tampering			X	X
Repudiation			X	
Information Disclosure			X	X
Denial of Service	X	X	X	X
Elevation of Privilege	X	X	X	X

The symbol X denotes the existence of a STRIDE component.

The different deployment models of the cloud provide sharing of user data to more than one operating organisation, sometimes for Personal Identity Information (PII) authorisation. Organisations must maintain information confidentiality and integrity to avoid data tampering by unauthorised users in communicating data across boundaries. Only data encryption will not be enough to ensure integrity. Furthermore, for PaaS and SaaS static data, only encryption mechanisms might not be enough to ensure perfect forward secrecy: the assurance that the system will always maintain data security in the advent of a breach. Indexing and querying of static encrypted cloud data can primarily be accomplished through searchable encryption, which is only exponentially efficient [43]. As a result, most static data for cloud-based applications are generally unencrypted, which poses a considerable threat to data security in the cloud.

3. Cloud Computing Security

Cloud computing's diverse range of applications has drawn academic attention to security when it comes to data storing, management and processing [44]. Cloud computing brings open issues regarding the security and privacy of outsourced data. Due to its dynamic abstraction and scalability, applications and data outsourced to the cloud have unlimited security boundaries and infrastructure. Another primary security concern surrounding the adoption of cloud computing is its multi-tenancy nature and sharing of virtualised resources [10]. Cloud providers such as Google, Microsoft, and Amazon have recently accelerated their cloud computing infrastructure and services to support a more considerable amount of users [39]. Nevertheless, the issue of privacy and security will continue to grow because cloud databases usually contain important sensitive information [45]. The confidence level in adopting the cloud is dropping due to the threats analysed in Table 7 and highlighted as follows [46].

1. Immoral use and abuse of cloud computing: Cloud computing infrastructure offers various utilities for users, including storage and bandwidth capacities. However, the cloud infrastructure lacks full control over the use of these resources, granting malicious users and attackers the zeal to exploit these weaknesses. Malicious users abuse cloud resources by targeting attack points and launching DDoS, Captcha solving farms and password cracking attacks. These threats mostly affect the PaaS and IaaS layers due to their high user interaction level.
2. Malicious insider attackers: Attacks generated from malicious insiders have been one of the most neglected attacks, but it has been the most devastating form of attack affecting all layers of the cloud infrastructure. A malicious insider with high-level access can gain root privilege to network components, tampering with sensitive and confidential data. This attack poses many security threats because Intrusion Detection Systems [47] and firewalls bypass such anomalous behaviours, assuming it as a legal activity, thereby posing no risk of detection.
3. Vulnerable programming interfaces: Part of the cloud services for user interaction in all layers is publishing APIs for easy deployment or the development of software applications. These interfaces provide an extra layer to the cloud framework to increase complexity. Unfortunately, these interfaces bring vulnerabilities in the APIs

for malicious users to exploit through backdoor access. These types of vulnerabilities can affect the underlying operations of the cloud architecture.

4. Data leakage and loss: One of the significant concerns of cloud computing is data leakage due to the constant migration and transmission of information over untrusted channels [10]. Loss of data can lead to data theft, which has become the biggest threat to the IT world, costing clients and industries a massive amount of money in losses. Causes of data loss result from weak authentication and encryption schemes, defective data centres, and a lack of disaster control.
5. Distributed technology vulnerabilities: The multi-tenant architecture offers virtualisation for shared on-demand services, meaning that one application can be shared among several users, as long as they have access. However, vulnerabilities in the hypervisor allow malicious intruders to gain control over legitimate virtual machines. These vulnerabilities can also affect the underlying operations of the cloud architecture, thereby altering its regular operation.
6. Services and account hijacking: This is the ability of a malicious intruder to redirect a web service to an illegitimate website. Malicious intruders then have access to the legitimate site and reused credentials and perform phishing attacks and identity theft.
7. Anonymous profile threat: cloud services possess the ability to provide less involvement and maintenance for hardware and software. However, this poses threats to security compliance, hardening, auditing, patching, logging processes and lack of awareness of internal security measures. An anonymous profile threat can expose an organisation to the significant risk of confidential information disclosure.

Table 7. Cloud computing security vulnerabilities using STRIDE.

Vulnerability Component	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Immoral use and abuse of cloud computing		X	X	X	X	
Malicious insider attackers	X	X	X	X	X	X
Vulnerable programming interfaces		X		X		X
Data leakage and loss		X	X	X	X	
Distributed technology vulnerabilities	X	X				X
Services and account hijacking	X	X	X	X	X	X
Anonymous profile threat		X	X	X	X	

The symbol X denotes the existence of a STRIDE component.

The distributed and shared nature of the underlying cloud infrastructure has made it challenging to design a self-security model for ensuring adequate data privacy and security. Adversaries exploit these security concerns in cloud architecture using sophisticated techniques to gain privilege or root access into the network. The Internet Protocols pose vulnerabilities for attacking cloud systems, such as man-in-the-middle, ARP spoofing, DNS poisoning and IP spoofing. A summary of these attacks is shown in Table 8. ARP poisoning is one of the principal vulnerabilities in the IP protocol stack. Exploiting this vulnerability, malicious users can redirect outbound and inbound traffic of legitimate users since the Address Resolution Protocol does not often require any Proof. Web services (HTTP protocol) session states and many techniques have been known in the literature for exploiting session handling, such as session hijacking and ridding. Injection attack vulnerabilities, such as operating system and SQL injection flaws, are used to divulge application modules. These application modules can represent the core of organisational data containing sensitive private information stored in the cloud. Availability and the functional operation of the cloud sometimes depend on how to secure the provided APIs. Insecure APIs can lead to HTML service attacks, such as browser phishing, and malicious users can launch SSL certificate spoofing.

DoS/DDoS attacks affect the security of cloud services. DDoS attacks launched on a system can disrupt the Quality of Service and legitimate user access. Intrusion Detection Systems (IDS) are adopted in preventing DDoS attacks. The goal of an IDS is

to feed in an extra layer of defence or protection against malicious users exploiting the vulnerabilities of computing systems by alerting users of any abnormal behaviour. IDSs are essential in detecting cloud service disruptions [37]. Table 9 provides a summary of possible Intrusion Detection attacks. Cloud security is dependent primarily on SaaS layers and web applications because they mainly offer cloud services. Therefore, the availability and security of the overall cloud services are dependent on the overall safety of the APIs, software applications, and web browsers [17].

Table 8. Common security attacks.

Classification of Attack	Description	Attack Name
Denial of Service	Large amount of data traffic is generated by the attacker to obstruct the availability of services	SMURF: ICMP: generating echo request to an intending IP address. LAND: transferring spoofed SYN packets with the same source and destination IP address. SYN Flood: reducing storage efficiency through IP spoofed packets. Teardrop: exploiting flaw TCP/IP stacks.
Distributed Denial of Service	A DDoS is the distributed form of DoS where the system is flooded in a distributed manner.	HTTP Flooding: exploiting legitimate HTTP POST or GET requests. Zero Day Attacks: exploiting security loopholes unknown to CSPs.
Remote to Local	Attacker compromises the system by executing commands that grants access to the system.	SPY: installations that runs a machine for phishing purposes. Password Guess. IMAP: finding a vulnerable IMAP Mail server.
User to Root	Attacker gains root access to destroy the system.	Rootkits: Offering privileged access while masking its existence. Buffer Overflowing
Probing	Breaching the PII of a victim	Ports Sweeping. NMAP: port scanning.

Table 9. Intrusion detection security threats to cloud computing.

Attack Name	Description	Affected Layer
Service Injection	This attack affects the integrity of services at the application and VM level. This is accomplished through the injection of malicious services into legitimate identification files. This, in turn, provides malicious services instead of legal services.	PaaS
Zombie	Impedes on availability of service by compromising legitimate VMs through direct or indirect host machine flooding.	PaaS, IaaS and Saas
Hypervisor and VM Attack	By compromising the hypervisor, the intruder gains access to a users VM, through the escape of a virtualisation layer.	IaaS
Man in the Middle	Accessing data transfer or communication to users. These affect the integrity and confidentiality of the message.	PaaS, IaaS and Saas
Back Door Channel	This attack affects the data privacy and availability of service. This is accomplished by the compromise of a valid VM, by providing rights to access resources.	Iaas

Table 9. *Cont.*

Attack Name	Description	Affected Layer
Phishing	Making users access fake or illegal web links. This can affect the privacy of user sensitive data.	PaaS, IaaS and Saas
Spoofing Meta Data	This affects the confidentiality of services through service abnormal behaviours by modifying the web service description.	PaaS and SaaS
Side Channel Attack	This affects data integrity. Hackers are able to retrieve plaintext or cyphertext from encrypted data through side channel information. These can be performed either through unauthorised placement of the effected text on users VM or through target VN extraction.	SaaS and PaaS
Authentication Attack	Exploiting flaws in the authentication protocol.	PaaS, IaaS and SaaS

3.1. User-Centric Cloud Accountability

Cloud accountability provides mechanisms and tools that help achieve responsibility and trust from cloud providers to users. Unauthorised data access is a primary factor in checking whether CSPs observe the legal agreement of providing accountability or if collected data are processed correctly. Users should have the user-centric ability to inform CSPs of possible harmful behaviours and errors. For instance, the right to obtain access contributes to both parties' accountability, also building an asymmetric power relationship between users and CSPs. Definitions 2 and 3 provides the logical definition and the requirement of CSP for cloud accountability [40].

Definition 2. *Cloud Service Accountability (CSA): A cloud service is said to be accountable if* $CSA = \{CP, sc, CI_{id}, i\}$*, where service contract* $sc = \{sow, R, sla, T, P\}$*,* $sow = \{F_p, O_p, F_c, O_c\}$ *as the statement of work,* $sla = \{F_p, O_p\}$ *as the service level agreement, P represents the parties involved* $P = \{s_p, s_c\}$*. R is the set of rules and* $T = \{start - time, end - time\}$*.* F_p *represents a set of provider's prohibited clauses,* O_p *is the provider's obligations set,* F_c *is set of consumer forbidden clauses and* O_c *consumer obligations set.*

Definition 3. *General Checking of Accountability Breach:*

If we denote s to be a service and its corresponding invariant to be $V = v_1, v_2, v_3 \cdots v_k$*. let the critical to accountability preconditions of s be* $PR = \{pr_1, pr_2, \cdots, pr_j\}$ *and the post-conditions be* $PO = \{po_1, po_2, \ldots, po_k\}$*.*
GIVEN: *Service providers and users fulfil all sets of n preconditions,*
$\forall pr \in PR, pr = true$*;*
CHECK: *Making sure that there is no invalidation to any of the invariants and post-conditions,*
$\forall po \in PO,$ ***ASSERT***$(po = true)$*;*
$\forall v \in V$ *,* ***ASSERT***$(v = true)$*;*
At any instance, the ***CHECK*** *fails,*
then there has been a breach in trust and accountability

Ko et al. proposed the TrustCloud framework that implements abstraction layers independent of one another. The TrustCloud framework was more of a detective approach than a preventive approach, arguing that detective methodologies can supplement preventive methods because they are non-invasive, investigating external risk and risks that can arise from CSPs. Li et al. [48] proposed a mechanism for trust in the cloud using a multi-tenancy trusted computing environment model (MTCEM). The model was designed to help with the duty separation in security between the CSPs and customers. The developed model was for the IaaS service infrastructure model, whose responsibility was to separate the security responsibility of cloud infrastructures. The model was made of Platform Configuration

Registers (PCR) that can prevent both the history of recorded information in an orderly fashion and event-related information. The general purpose of the model was to assure that CSPs will play their security role by securing the infrastructures and that customers must build trusted virtual instances for themselves. In the sense that no parties involved in the communication process will violate each other's authority. In this case, building trust in one another.

In the works of Carmen et al., they defined tools that facilitated the appropriate choice for CSPs described as Cloud Offerings Advisory Tool (COAT) and Data Protection Impact Assessment Tool (DPIAT). These tools were designed to take charge of data control of users, such as Data Track (DT), and tools that will help implement accountability and specify related policies using Accountability-Primelife Policy Language (A-PPL) and associated enforcement engines. The aftermath of implementing the COAT was designed to be an immediate and sustainable changeable response panel that comprises an overview of the companionable package offerings, corresponding to the user's requirement by linking its informational analysis. In accomplishing this, a familiar store-type interface was used to reduce complex issues and increase usability. The authors' main goal was to design and implement an A4Cloud framework that provides trustworthy requirements in cloud services by devising tools and methodologies that cloud stakeholders would be held accountable for violating confidentiality. Open issues arising from accountability in the cloud, which serve as counterexamples and limitations to [3], are the lack of automation for cloud accountability projects for users and issues that arise from integrity, which may occur on parts of the cloud providers [29]. Other limitations are hidden identity violation, secure provenance, and collaborative monitoring. Matters arising from the implementation of works in [48] are the issue of a non-bargained untrusted CSP: failure of CSPs effectively playing their role.

3.2. Digital Identity Management

Digital Identity Management (IdM) is a crucial feature in cloud computing infrastructures for supporting adaptable access control and authenticating users based on their identity properties, attributes, or past interactions [49]. An essential requirement for developing digital identity management systems suitable for cloud computing is the ability for cloud users to have control over their PII to reduce identity theft or fraud. In cloud IdM platforms, there is the issue of interoperability, which ranges from using several identity tokens, such as the encodings in SAML assertions and X.509 certificates, and distinctive identity intervention procedures such as identity-provider centric and client-centric protocols, to the use of identity attributes. An identity attribute encodes a piece of specific identifying information about an individual, such as the social-security-number; it consists of an attribute name, also called identity tag, and a value.

From the literature, basic techniques for Identity Management by [49] provided simple architecture that implements Zero-knowledge Proof Protocols and semantic matching techniques. The authors further added an extension to the protocol by introducing Aggregated Zero-Knowledge Proof of Knowledge. The enhancement by [49] provided cryptographic features that allowed clients to verify the information of many identity attributes using a single interaction, without the essence of offering them plain sight by computing the client's commitment as $M = \prod_{n}^{i=1} M_i = g^{m_1+m_2+m_3\cdots+m_i} h^{r_1+r_2+r_3\cdots+r_i}$ with corresponding signatures $\sigma = \prod_{i=1}^{n} \sigma_i$, where σ_i is the signature on $M_i = g^{m_i} h^{r_i}$.

Paci [50] proposed similar works to [49] but on mobile devices. The authors in [50] designed a system called VeryIDX. The system demonstrated privacy-preserving management of users' identities using identity attributes. The proposed VeryIDX architecture was composed of three modules: The Registrar (R), Service Provider (SP) Application, and a client application (C). The registrar must store clients' identity records (IdRs), representing users' identity attributes. Each IdR, in turn, consists of several identity tuples, one meant for each user's identity attribute. An identity tuple then stores the message m and the registrar's signature on the commitment of m, denoted as T_i. Overall, the VeryIDX was

centred on the concept of multi-factor privacy-preserving verification of identity attributes achieved using an aggregated ZKPK protocol, as in [49].

Issues arising from [49,50] are the problems of scalability and user accountability. There is always a trade-off between integrity and accountability, which holds a thin line when preserving privacy. In this case, accountability has greater weight than the integrity of the user. These designed systems jeopardise the overall security. A further issue is the problem of complete controllability [29]. Fully dynamic ability support from works from the literature causes higher computational, communication, and storage overhead. In the aspect of accountability for IdM systems, Priem et al. and Paci et al. both emphasised that for accountability of an individual, IdM scheme's occasionally disclosed identities or credentials must be entrusted to ensure that an actor meets the privacy demands. However, a significant factor to be considered for accountability when designing privacy-enhanced IdM schemes is that individuals should be able to trust that only accountability requirements will be sufficient to preserve their privacy.

3.3. Data Integrity

Zero-Knowledge Proofs [51] are mechanisms that allow two party's to prove to each other that a given statement is true without revealing further information that will jeopardise the integrity of the other, shown in Definition 4.

Definition 4. *Let $PK(x) : y = f(x)$ denote the "zero-knowledge-proof-of-knowledge. Given that x = secret input, such that $y = f(x)$" In technical terms: there is the existence of a knowledge extractor that will extract x from the prover with negligible probability of the information on x. Most importantly, the prover with no knowledge of x convinces the verifier with negligible probability.*

A more practical application is zero-knowledge Succinct Non-interactive ARgument of Knowledge (zkSNARKs): an architecture that enables users to broadcast proven encrypted information without disclosing the contents. zkSNARK allows a prover P to convince a verifier V of a statement of the form "given a function F and input x; there is a secret w such that $F(x; w) = true$", preserving the privacy and integrity of the participants involved in a transaction (for instance: a user and a server). The concepts of the zkSNARKs come with a considerable cost, which is the high computing overhead of V, as a result of the monolithic architecture: functionally different aspects (for example, data input and output, data processing, error handling, and the user interface) are all interwoven, rather than containing architecturally separate components, which play a massive role in the computing power of the system [52]). Wu et al. proposed a Distributed Zero Knowledge Proof (DIZK) that distributes the generation of a zero-knowledge proof across machines in a compute cluster. The server S and prover P in DIZK were modified from a monolithic architecture to a distributed manner of clusters. One significant advantage of this architecture is implementing cloud platforms to prove the integrity of either the server or a third-party involved in the computation, especially in health care environments where patient sensitive data are stored in the cloud.

Another way to ensure security is the introduction of *Auditabilty* by a third-party for CSPs. Authors in [53] have argued that resorting to public auditability for data outsourced in the cloud is of crucial importance. Users can resort to an external auditor to check the integrity of outsourced data when needed—an architecture for data integrity using an external auditor. The question is, how can a Third-Party Auditor (TPA) efficiently and effectively audit the data outsourced in the cloud without introducing any additional burden on a cloud user and not demanding a local copy of the data storage? Secondly, how can the introduction of TPAs not bring any unforeseen vulnerabilities that will hamper users' privacy? Wang et al. [53] proposed an approach that utilised the combination of random masking with a public key-based homomorphic authenticator: enabling a client to authenticate a large collection of data elements $m_1, \ldots, m_t$ and outsource them along with the corresponding authenticators to an untrusted server. At any later point, the server

can generate a short authenticator $\rho f, y$ vouching for the correctness of the output y of a function f computed on the outsourced data as $y = f(m_1, \ldots, m_t)$ [54]. The use of HAs in [53,54] is that they can generate unforgeable verification of metadata from any individual data block. This assures effective and secure aggregation of data blocks to be correctly computed by verifying an aggregated authenticator. With the random masking technique, TPAs no longer have access to all necessary data to develop a correct set of linear equations for data modification. Therefore, TPAs cannot derive the content of users' data no matter the number of the same linear combinations of file blocks generated. Neenu et al. [55] proposed index-based time stamps for data stored in the Markle hash tree. Their proposed scheme was computationally effective for dynamic data stored in the cloud. The limitations of works in the literature such as [55–62] are the assumption of a trusted auditor, which is not efficient in a real-life scenario, and the issue of a single point of failure for the case of a single trusted auditor. A better approach will be to implement a decentralised scheme for a third-party auditor to eliminate the issue of a single point of failure and trust. Authors in [63–66] proposed the use of blockchain for data integrity. Unfortunately, there is a computational issue for proof of work.

3.4. Cloud Intrusion and Detection

There exist different methods in detecting and preventing intrusion and detection attacks in the cloud, such as statistical analysis, data mining and machine learning algorithms. Statistical analysis detection methods detect anomaly behaviours through computational analysis of the network. The advantage of this approach is the lack of training or prior understanding of the security risks involved in the network traffic. Its limitation is its inefficiency in detecting anomaly behaviours due to incomplete knowledge [67]. The data mining method uses the concept of classification, association, and clustering rules to detect anomalies. This approach can be very flexible and easy to deploy. Still, its disadvantage is its inaccuracy in parameter manipulation, which can alter privacy settings and security protocols. Machine learning algorithms can improve computational performance by learning or training parameters involved in the computation. This approach can create a system that will enhance the performance of a program through a learning process that will improve on previous results. Another exciting feature of this approach is its ability to learn more information from previous results to improve future performance.

The introduction of machine learning as a control for Intrusion Detection has made the IDS implementation efficient and scalable. The network can be correctly trained to detect intrusions. For a given intrusion sample set, the network learns to identify behavioural patterns in the model. With an extensive training dataset of attacks, the network can identify a broader range of unknown attacks. Machine learning algorithms such as Support Vector Machines, Artificial Neural Network, Data Mining and Fuzzy logic have been adopted in cloud computing for intrusion detection attacks. The use of ML algorithms drives solutions to the problem of analysing massive data network traffic and realising better performance optimisation for detection [68]. The machine learning classifiers are stated below.

1. Decision Tree Algorithm: This technique is implemented through the concept of game theory. The DT algorithm is implemented in Intrusion Detection Systems by choosing splitting attributes with the highest information gain using Equation (1), because the probability of occurrence of an attribute is based on the amount of information that can be associated with the attribute. Let the D and $H(D)$ be the data in a given dataset, and C be the associated class, then

$$Gain(D, S) = H(D) - \sum_{i=1}^{S} p(D_i)H(D_i) \quad (1)$$

Quantifying the information gain of an attribute is achieved through the concept of entropy by measuring the level of randomness in a dataset, as shown in Equation (2).

If the data belongs to a single dataset with no uncertainty, then the entropy is zero, as established in Equation (2).

$$Entropy : H(p_1, p_2, \ldots, p_s) = \sum_{i=1}^{S}(p_i[log(1/p_i)]) \quad (2)$$

One main advantage of the DT classifier is that it constantly partitions the given dataset into subsets for all elements, where final subsets belong to the same class.

2. K-Nearest Neighbour (KNN): The KNN algorithm is based on distance measures between classes. It seeks to find k attributes in the training data, which seem to be closest to the test example [68]. After which, it assigns the most frequent label among these examples to the new model. Whenever any classification is made, it first calculates its distance to each attribute contained in the dataset and only k closest ones are considered.
3. Bayes Rule (BR): BR calculates the probability of a hypothesis based on prior probability, as depicted in Equation (3). Given an observed dataset D and any form of initial knowledge, the best possible hypothesis will be the most probable one. Given that $h = hypothesis$, $P(h|D) = posteriorprobability$, $p(h) = priorprobability$. In some cases where we are most interested in calculating the most probable hypothesis ($h \epsilon H$), this is defined as the Maximum Posterior Hypothesis (MPH), defined in Equation (4). From Equation (4), if we assume that the probability of the data $P(D)$ is constant because of its dependency on the hypothesis h, then $P(D|h)$ is called the Maximum Likelihood (ML) hypothesis, shown in Equation (5).

$$BR : P(h|D) = \frac{P(D|h)P(h)}{P(D)} \quad (3)$$

$$h_{mps} \equiv argmax_{h \epsilon H} P(h|D) \quad (4)$$

$$= argmax_{h \epsilon H} \frac{P(h|D)P(h)}{P(D)} = argmax_{h \epsilon H} P(D|h)P(h)$$

$$h_{ml} \equiv argmax_{h \epsilon H} P(D|h) \quad (5)$$

4. Naive Bayesian (NB): NB is a probabilistic approach very similar to the Bayesian Rule. It computes the probability of each class and then determines which attributes to classify and learn to predict the new class. Given a vector V represented by n different variables $V = \{V_1, V_2, V_3 \ldots V_n\}$ assigned to probability instances $P = \{Ck|V_1, V_2, V_3 \ldots V_n\}$ for every k possible results or classes Ck, the conditional probability can be formulated, as shown in Equation (6).

$$P(Ck|V) = \frac{P(V|Ck)P(Ck)}{P(V)} \quad (6)$$

where $P(Ck|V)$ = Posterior Probability, $P(V|Ck) = PriorProbability$, $P(Ck)$ = Likelihood and $P(V)$ = Evidence. The joint computation can then be written as follows

$$P(Ck) = \prod_{i=1}^{n} P(v_i|Ck) \quad (7)$$

5. Support Vector Machines (SVM): SVM is a numerical learning model centred on a data-mining approach. It was initially introduced for only data classification, but with the advance of complex situations, it has now been fully implemented for clustering tasks and regression analysis. There are different notions about the performance level

of SVM compared to neural networks. Still, many authors from the literature agree that SVM performs better than the multi-layer perceptron as a result of its reversed neural network design [69]. The SVM can also be used in spam filtering pattern recognition and anomaly network detection [70]. Training data usually achieve the near precise SVM classification to classify unidentified samples given training model data. SVM has the advantage of finding an optimum global result by performing linear separation in a hyperplane to two separate classes. After this separation, the closest data to the hyperplane are classified as the correct class. Considering a training dataset $D_l = \{x_i, y_i\}_{i=l}^{l}$, $x_i = i_{th}$ input vector for $x_i \epsilon R^n$, $y_i \epsilon [+1, -1]$, where l = total number of input vectors, and n = dimension of the input vector space. Assuming the relationship between x and y be $y = Sgn(f(x) + \epsilon)$, where $Sgn(x) = i$ if $x \geq 0$ and $Sgn(x) = i$ if $x < 0$. Then, the task to uncover f is called the *Classification Function*. SVM evaluates Equation (8) to create a trade-off between complexity and empirical error of the hypothesis space, where C = the regularisation parameter that will control the identified trade-offs of the used hypothesis space.

$$\min_f \|f\|_k^2 + C \sum_{i=1}^{l} |1 - y_i f(\mathbf{X}_i)| \tag{8}$$

Providing security measures for distributed models such as cloud environment entails more than just passwords for user authentication or digital certificates for confidentiality when transmitting information [71]. The distributed model nature of the cloud has made it highly vulnerable and prone to sophisticated distributed intrusion attacks such as Cross-Site Scripting (XSS) and Distributed Denial of Service (DDOS). Before the widespread use of machine learning applications. Traditional IP and packet filtering approaches were implemented to handle significant network control over accessed traffic. Authors such as [71] proposed a multi-threaded distributed cloud IDS to mitigate against large data flow of packets, analyse the packet and efficiently generate reports by integrating knowledge and behaviour analysis to detect intrusions. The multi-threaded architecture was monitored and administered by a third-party monitoring service in their implemented mechanism. The third-party monitoring service then generates alerts and mitigation control for CSPs. The proposed model was designed using three dependent modules: capture, analysis and reporting modules. These three modules identified an efficient matching and analysis of bad packets and CSP-generated alerts. The proposed model's strength was that the multi-threaded approach could handle a large volume of data in the cloud. Secondly, the cloud IDS improved efficiency due to the reduced memory, CPU consumption, and packet loss. A limitation to the proposed model was the introduction of third-party control over the multi-threaded approach. The issue of high compromise can arise because of the single point of failure and bottleneck.

The use of ML algorithms drives solutions to the problems of analysing huge data network traffic and realising better performance optimisation for detection [68]. Farid et al. [68] proposed an Improved Self Adaptive Bayesian Algorithm (ISABA) for cloud-based intrusion detection. The Adaptive Bayesian Algorithm generates a function from the training dataset. This function estimates the conditional class probabilities for each attribute based on their frequencies over the weights, putting a match of the same class in the same training dataset. For improvement to this (ISABA), given any intrusion training data, the weights are initialised for each W_i set to 1.0. Then, the prior probability is estimated by summing the weights of how often each class occurs in the training data. When there is misclassification in training, the prior and conditional probabilities are recomputed using the training examples, and then the weights are updated. The continuous iteration of these processes achieved target accuracy. Experimental results proved that the ISABA outperformed the SVM, NB and NN for training and testing and classification rates. Further research on the ISABA will apply domain knowledge security in improving its detection accuracy. Wani et al. [72] proposed an intrusion detection system that was tested on three different

ML algorithms, namely: SVM, naive Bayes and random forest. Experimental analysis was carried out for accuracy, recall and precision on normal packets and DDoS packets. After proper training using the three ML algorithms, SVM depicted greater precision and accuracy than the other classification algorithms. The limitation was the inability to detect some well-known Intrusion Detection attacks, such as zero-day attacks and zombie attacks. Bhamare et al. [73] argued that it is relevant to test an ID system with a different dataset to create effective detection attacks. Therefore, the authors proposed training supervised ML algorithms such as SVM, logistic regression, decision tree and naive Bayes with two different datasets, UNSW and ISOT, using the WEKA tool. At the end of the experiment, it was found out that SVM and decision tree both averagely performed better than the remaining algorithms. The claims of Bhamare et al. were right in the sense that one particular algorithm did not outperform the remaining algorithms in both datasets, implying that there is an imbalance between supervised learning algorithms. This may be because these algorithms perform better with a large number of negative and positive samples. The works of [73] create new ideas for training models for detection, which means there is a need to train a model with multiple datasets before validating the efficiency of that model.

Rodrigues [74] proposed an NN model that is trained based on users timing vectors from keystroke properties extracted from users inputs login name and password strokes. After the training, when a login name and password is entered, the user's timing vector is applied to the NN. The resulting input–output difference is compared using the predetermined threshold, and access is denied at any instance where the difference is larger than the threshold. Osanaiye et al. [75] proposed an ensemble-based method that implements the multi-filter feature selection to combine the output results of four different filter methods to achieve maximum results. According to their argument, there are three trends in literature for feature selection: identity correlation, unique identity features, and robust but individually weak features. After implementing these features, performance measures such as information gain and gain ratio were conducted, and the approach performed better than the traditional SVM approach. This means that it is always a better choice to integrate feature selection than a single feature implementation. Gill et al. [76] proposed a self-protection approach in cloud resource management called SECURE. SECURE was capable of automatically generating signatures to mitigate attacks. SECURE adopted SVM as a security agent to detect anomalies in network traffic. These anomalies were stored in the database for future comparison. The approach could self-protect from intruders by differentiating illegitimate and legitimate behaviour. The approach's strengths were the ability to detect attacks while processing workload continuously. The limitation to SECURE was the inability to efficiently detect zero-day attacks, which can be improved by locating the source of the attack using learned behavioural patterns.

4. Privacy Preserving in Cloud Computing

To preserve privacy and reduce the level of distraction conflicting with users' privacy in the cloud, there's a need to provide privacy-preserving protocols that maintain the confidentiality of the user [77]. Definition 5 provides a full description of what it means to be Privacy-preserving.

Definition 5. *Privacy-preserving: Let i be an instance from site S with a attributes, and a_i denoting an i_{th} attribute of a. If we also assume some set of rules $r \in R$ provided by another site S' for each attribute in the form of $(N_1 \wedge N_2 \wedge \cdots \wedge N_v) \rightarrow C$, where C is the predicted class if $(N_1 \wedge N_2 \wedge \cdots \wedge N_v)$ is true. In addition, if S has a set E of rules that have not been used in the classification, then the system is said to be privacy-preserving if no party can gain extra information about the number of clauses in a rule, such as:*

1. *S will not be able to learn any rules in R.*
2. *S will be convinced that $E \cap R = \varphi$ holds.*
3. *S' will only learn the class value of a and what is implied by the class value.*

There are three main ways to achieve privacy. They are namely:

- Privacy-Preserving Additive Splitting Technique: If a value x is assumed as input, then x is said to be additively split between different parties A and B, if A has a random x_A and B has a random x_B, such that $x_A + x_B = x$, where the addition is modular. If y is split in a similar manner $(= y_A + y_B)$ then A and B can compute the sum of x and y by adding their respective shares of x and y, that is, if $z = x + y$, then A computes $z_A = x_A + y_A$ and B computes $z_B = x_B + y_B$. Computing $z = x * y$ in split form is considerably complicated if x and y are additively split.
- Privacy-Preserving Encoding Based Splitting Technique: This is the process where only A generates an encoding known to only A, and another party B computes the encoded element but has no meaning to B. In other words, B does not know what the encoding of A means. As an example, let i represent an intermediary Boolean variable. If A generates a random value $r_i[0]$ as the encoding for i, and another randomly generated value $r_i[1]$ for encoding the value 1. As the computation proceeds, B is able to see the encodings $r_i[0]$ or $r_i[1]$ but cannot deduce their meaning.
- Homomorphic Encryption: Using homomorphic encryption, a cryptosystem E is said to be homomorphic in message space M and ciphertext C such that $\forall m_1, m_2 \, \epsilon \, M : E(m_1 \odot_M m_2) = E(m_1) \odot_c E(m_2)$. Where $\odot_M$ and $\odot_c$ are the binary operators in $plaintext : M$ and $Ciphertext : C$. If we denote an encryption function by E_{pk} and a decryption function by D_{sk}, then it is possible to compute $E_{pk}(x + y)$ of two inputs x and y that are encrypted as $E_{pk}(x)$ and $E_{pk}(y)$ by computing $E_{pk}(x) * E_{pk}(y)$. Furthermore, with $E_{pk}(x)$, it is possible to compute $E_{pk}(c * x)$ for any constant c by computing $E_{pk}(x)c$.

According to Definition 5, cloud computing protocols are said to be privacy-preserving if only what it reveals is because of a collaboration that is deduced given the participant's input set [78]. Due to the multi-tenant nature of the cloud, security attributes and policies may directly or indirectly affect privacy-preservability, which can be in the form of integrity, accountability or confidentiality [29]. User confidentiality can become indispensable when maintaining the nondisclosure of private data, and integrity will ensure that computational data are not corrupted, which is privacy-preserving. On the other hand, accountability might undermine privacy due to the conflicts in achieving the two attributes. Therefore, privacy-preservability can be defined as a stricter form of confidentiality because they both prevent information leakage. This infers that violating cloud confidentiality will also break privacy-preservability.

4.1. Data Privacy

It is well-known that storage data encryption is not fully efficient in preserving the privacy of outsourced storage applications [79]. The frequency of accessed storage locations from the server by users can leak a substantial volume of sensitive user information through statistical interpretation for unencrypted data [43]. Since only cryptographic techniques cannot ensure privacy, Goldreich and Ostrovsky [80] first proposed the concept of Oblivious RAM using Definition 6.

Definition 6. *A data access is said to be oblivious if accessing the cells of A according to a random hash function, $h(i)$, as $A[h(1)], A[h(2)], :::, A[h(n)]$, or random permutation, $\pi(i)$ as $A[\pi(1)], A[\pi(2)], :::, A[\pi(n)]$, and not oblivious if $T[h(A[1])], T[h(A[2])], :::, T[h(A[n])]$, where T is a hash table*

The method employed by Oblivious RAM allows a client to conceal its access pattern to the remote storage by continuous shuffle and data re-encryption as they are accessed. Even if a malicious attacker observes or intercepts storage locations, the Oblivious RAM algorithm ensures that the adversary has a negligible probability of learning anything about the true logical access pattern. To further enhance user access patterns when using ORAMs, Goodrich et al. [81] proposed privacy-preserving data access using a combination

of probabilistic encryption, which directly hides data values, and stateless oblivious RAM simulation, which hides the pattern of data accesses. The limitation to this was the worst-case efficiency of the algorithm, which achieved $O(\log n)$ amortised access overhead. Further works by Stefanov et al. [82] achieved $O(\log^2 N/\log \chi)$, for large block size $B = \chi logN$ with a reduction rate of $\chi \geq 2$ for every $N' = N/\chi$, where N represents the number of blocks. The integrity check of the Path ORAM was based on the concept of the Markle Tree, where data storage is placed at every node of the tree and not only the leaf nodes. Tagging every node bucket of the Path ORAM with a hash of the form $H(b_1||b_2 ::: ||b_Z||h_1||h_2)$, where b_i for $i \epsilon \{1, 2, 3, 4, 5, \ldots, Z\}$ represents bucket blocks, and h_1 and h_2 represent the left and right leaf hashes, respectively. Therefore, for leaf nodes, $h_1 = h_2 = 0$, only two nodes for each ReadBucket or WriteBucket operation will need to be read or written. However, Haider et al. [83] have shown that information can still be leaked even if only Write Access Patterns are visible to a malicious intruder. Instead, the authors [83] proposed the Flat ORAM. The algorithm comprises two zero-initialised OccMAp and PosMap corresponding to the occupancy and the position map of N and P entries, respectively, allocated. Then, each block is mapped to a random block, assuming that PosMap and OccMap both reside where the adversary has no access (on-chip). Any collision is being avoided with OccMap. The OccMap is then updated as 'occupied' for all assigned logical blocks. For an integrity check for Flat ORAM, let us assume a logical block with counter c, upon any writes, the controller computes MAC $h = MAC_K(a||c||data)$ using K as a secrete key, then writes the tuple $(h, data)$ to Dynamic RAM (DRAM). The hypothetically altered data tuple $(h*, data*)$ is read upon every read. Then the ORAM controller recomputes hash MAC $h = MAC_K(a||c||data*)$ to check if $h = h*$. If they are both equal, then data integrity has been verified. The mechanisms of the Flat ORAM shuffles only Write Access Patterns to conserve user privacy. Interestingly, it is preferred to a fully functional ORAM because it offers better performance and higher energy efficiency.

4.2. Access Control

User privacy concerns are not only dependent on what matters to a user of a system but depend on whether malicious intruders can have equal access [7]. Therefore, privacy preservation through access control comes from three major aspects, namely:

1. Information-Centric Security: Data objects should contain access-control policies. This can be implemented through outsourcing data architectures that integrate cryptographic techniques with access control [84].
2. Trusted Computing: Trusted cloud computing system that provides consistency in accordance with software or hardware specification [82].
3. Cryptographic Protocols: Cryptographic tools and techniques can be employed to achieve privacy, such as Fully Homomorphic Encryption (FHE) [85] and Attribute-Based Encryption [86].

Fall et al. [87] proposed a Risk adaptive Access Control (RAdAC) for preserving the privacy of sensitive data in flexible real-time. The RAdAC approach proposed by [87] was a hybrid of Policy-Based Access Control, Attribute-Based Access Control and Machine learning. In accomplishing this, the authors proposed principles to measure risk, establish an acceptable level of risk, and lastly, make sure that all information was tailored not to exceed the accepted level of risk. Following these guidelines, RAdAC provided adaptability and flexibility compared to the traditional access control. The RAdAC system discovered failures when access is being requested by checking past access control decisions and then quantifying the privileged, subject and object. Yu et al. [88] proposed a framework that models the way users interact to achieve goals. The proposed framework used a catalogue that guides software engineers through alternatives in achieving privacy. The author in [88] further shows ways for reasoning about the non-functional requirements for privacy by allowing only modelled relationships between users in a strategic manner. The framework's strength is its adaptive ability to achieve goals from different alternatives. Further enhancements to I* will be to study deeper the interrelationship between trust

and privacy, as this can interfere when allowing existing relationships. Kobsa et al. [89] proposed security requirements that guarantee privacy. The author also proposed ways to maintain user anonymity while preserving user privacy, implementing an architecture that provides security and privacy when using personalised cloud systems. The approach in [89] allowed users to hide their identities during data collection. Kobsa et al. used the concept of *Pseudoanonimity* because N entities will be unable to reveal user anonymity. To further defend anonymity, the users included one trusted entity in every $N+1$ component that may jeopardise anonymity. Coppolino et al. [90] proposed a solution in preserving the privacy of users through Homomorphic Encryption (HE) while detecting anomalous intrusion and detection in network traffic. The introduced HE scheme was used in encrypting the data from third-party monitoring services, which are intended to provide security. The architecture provided Adhoc Intrusion and detection for monitoring the third-party while preserving privacy. When considering attacks such as code injection techniques, the HE scheme implemented in [90] was able to monitor generated code injection attacks without accessing any unencrypted data files. The limitation is the excess overhead incurred from the HE scheme due to additional processing time for both encryption and decryption. Secondly, the HE scheme evaluation results are always ciphered and decrypted, creating a bottleneck for the IDS as it does not possess a private key.

4.3. Privacy Preservation through Access Patterns and Design

Privacy Process Patterns are specifically designed to model privacy issues effectively. They can be defined as patterns applied to privacy associated processes by specifying how privacy issues can be realised through identifiable procedures, connecting flows and the activities that link them. As supplementary, they assist software developers to understand how better to implement several privacy properties in a more precise manner. Privacy Process Patterns (PPP) are considered a more robust way to bridge the gap between user confidentiality and cloud service providers. Privacy Pattern Properties are defined as follows [91]

1. *Anonymity* can be defined as a quality that does not permit the user to be identified in any form, either directly or indirectly. A problem that can arise when a user is anonymous is the issue of *Accountability* and a large anonymity set. The benefits include location tracking freedom, users freedom of expression and low user involvement. This property can be implemented using Tor [92], Onion routing [93] and DC-nets [94]
2. *Pseudonymity* can be defined as the utilisation of an alias instead of personally identifiable information. A problem that can arise is the issue of *Integrity* [95]. The benefits include supporting user access to services without disclosing real identities. Users still maintain integrity protocol. This property can be implemented using administrative tools such as biometrics, identity management and smart cards.
3. *Unlinkability* can be defined as using a service or resource with the inability of third-party linkage between the user and the service. Issue: *Integrity and Accountability*. Benefits: privacy-preserving by not allowing malicious monitoring of user experience. Implementation: Onion routing, Tor and DC-nets.
4. *Undetectability* inability of third-party tracking amongst a set of possible users. Issues: undetectability strength is highly dependent on the size of the undetectability set. Benefits: preserve users' privacy without allowing detectability of service by malicious intruders. Secondly, attackers cannot adequately detect the existence of an exact Item of Interest (IOI), e.g., the use of steganography and watermarking. Implementation: smartcards and permission management, encryption methods such as mail and transaction encryption.
5. *Unobservability* inability to perceive the existence of a user amongst a set of potential users. Issue: dependent on the integrity level and anonymity set. Benefits: anonymity and undetectability enforcement per resources. Secondly, ensuring user experience without the connection and observability of a third-party. Implementation: smartcards and permission management. Anonymizer services such as Tor, Hordes and GAP.

The literature has identified the need to introduce a Privacy by Design (PbD) to support the need for sensitive and confidential information stored, shared and distributed at the digital level [81,82,96]. From the literature, works are still in progress to define privacy design patterns in cloud computing. Developing a privacy pattern language will further assist developers in building the gap between the design and implementation phase. However, despite the works presented in the literature, there is still a gap between privacy design and implementation. Authors in [96] implemented and provided Privacy Process Patterns by Design that can be used to bridge gaps highlighted in the literature. The authors demonstrated the practicality of the application through JavaScript Object Notation (JSON) in conjunction with the Privacy Safeguard (PriS) methodology and applied them to a real case study. Further implementation of privacy access patterns was implemented by [81–83]. The challenges of Privacy by Design were highlighted by Diamantopoulou et al. as a factor of design and implementation of policies to be established by software engineers, as they lack a standard definition of privacy requirements and policies. Secondly, the lack of proper policy requirement knowledge for correct implementation. Therefore, there is a need to propose a set of Privacy Process Patterns that enhances the detailed knowledge of cloud computing and a distinct coalition between cloud computing infrastructure and privacy requirements. The proper implementation helps support a privacy-aware technique in bridging the gap between user confidentiality and cloud service providers.

The authors of [96] successfully designed a set of privacy process patterns that can be used to bridge the gap between privacy design and implementation and their instantiation in several platforms without expertise or skill limitations. The authors argued that privacy should be controllable through access patterns and designs in that it allows secrecy preferences by a user. This helps users of the system to be flexible when divulging Personal Identifiable Information [97]. Papanikolaou et al. [98] carried out extensive surveys on how to automate legal and regulatory processes for the regulation and extraction of privacy rules. The idea is to apply a link policy and compliant techniques to provide salient means for maintaining and achieving user privacy in the cloud.

5. Final Remarks

From the review conducted, considerations were made based on cloud computing security and privacy issues that demand self-adaptiveness. The multiple security threats posed by the security issues are depicted in Table 10. Table 10 shows that there is a need for control mechanisms that provide hybrid mitigation when designing security implementation for cloud infrastructure. For instance, attack mitigation and control mechanisms such as ML algorithms for detection and prevention are faster and more accurate due to the high probability of detecting attacks compared to similar approaches using homomorphic encryption schemes. ML systems can recover from an integrity loss on time, gaining sufficient awareness without substantial availability loss. Therefore, knowing the damage of an attack campaign and how feasible it can become requires a high awareness level.

Table 10. Cloud computing security and privacy component using STRIDE.

Security Component	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Accountability		X		X		X
Identity Management	X		X	X	X	
Data Integrity		X	X	X		X
Intrusion and Detection	X	X		X	X	X
Data Privacy		X	X	X		X
Access Control	X	X	X		X	X
Access Patterns and Designs		X	X	X		

The symbol X denotes the existence of a STRIDE component.

5.1. Discussion

Cloud computing comprises heterogeneous resources at different geographical locations requiring a secure-aware approach to handling security threats. Private sensitive data belonging to distinctive organisations in community clouds should be separated to avoid intervening. This has become primarily impossible due to virtual machine image sharing among cloud providers [53]. Most present proactive mechanisms from the literature do not consider systems run time adaptation to either the authorisation infrastructure or service. That is, simultaneous detection or prevention in responding to abnormal behaviours: exceeded login attempts by malicious intruders when accessing a piece of information can be automatically tagged malicious without requiring any cryptographic scheme and human intervention. Furthermore, data-centric security information confidentiality should be preferred because it allows perfect forward secrecy, which is critical in mitigating against attackers eavesdropping or intercepting information in transit or at rest. Additionally, it minimises information security compromise and decreases the effect of the overall attack. Nevertheless, this approach will require techniques and protocols for originating temporary keys and the issue of regular updates. Without perfect forward secrecy, the confidentiality of data will depend on the efficiency of encryption keys, meaning that if private keys are leaked, packets might be decrypted when intercepted by an attacker. The inaccuracy in most developed systems results from the total dependency on cryptographic schemes. Because, in any case, where a loophole is found in the cryptographic scheme, then the entire system becomes invalid. Therefore, the evaluation of such cryptographic methods can be termed dangerous [97]. Authentication for cloud storage should not only depend on encryption schemes. Providing effective encryption schemes demands algorithms with significant key strengths. The limitations of large key strengths for cloud authentication are speed, processing power, and computational resources in encrypting a large amount of data. As key sizes increase, the maintenance and management of large key sizes become a bottleneck for the server. Cloud auditability also poses a significant issue, such as ascertaining the integrity of the data stored in the cloud without downloading the data first before uploading the data.

Cloud privacy protection and data security are primary issues for IaaS, Paas, and SaaS delivery models [76]. The security challenge is protecting privacy and PII while sharing data across different enterprises. Standard definitions of cloud policies, user confidentiality and integrity have not been adequately defined and therefore may conflict with each other, affecting the enforcement of confidentiality. Such conflicts have been depicted in previous implementations from literature, such as the complete anonymity to hide users' identities, which will make confidentiality and authentication more challenging. An extreme example is the situation of a shared file accessed by multiple users who may hide their identities due to anonymity for privacy protection, such as in the case of [82]. However, implementing such architectures, malicious users are hard to be tracked because of the user anonymity. Therefore, researchers must seek a trade-off in which the requirement of one attribute can be met while simultaneously maintaining a threshold degree of the other attribute.

The need to provide efficient security and privacy in the cloud is paramount [99]. Service providers need to control and guarantee users how their information is being accessed and what kind of information is released to the public. Secure-adaptive techniques should be implemented to enforce strict security in a cloud environment by providing separation between sensitive and non-sensitive data, followed by security mechanism such as encryption, privacy protection, and identity management frameworks. The current security mechanisms are incapable of providing a self-aware security approach from security attacks. Hackers and malicious intruders are very inventive when finding new ways to disrupt typical server and user operations. The introduction of adaptive systems will lower operation costs in complex changing environments and uncertainty by simultaneously adapting to the changes to achieve adequate security.

5.2. Conclusion

From the literature and trends of emerging technologies, the challenge in any system from the internet's critical infrastructures such as cloud computing is the ability of systems to self-protect regarding security and privacy. Secure adaptive techniques are ubiquitous and can be adopted at any stage of an underlining technology, from hardware and software to the core computing infrastructure. Secure adaptiveness implies that the system can self-protect during multiple attacks or a malicious user exploring multiple vulnerabilities. Cloud computing will still be prone to security and privacy concerns without the practical adoption of adaptive mechanisms for efficient client and user experience. This review highlighted the multiple vulnerabilities affecting the different components of cloud computing through STRIDE analysis. The study further provides limitations to different works from the literature, including classifying security and privacy issues based on attack mitigation. The review also provided a technical approach and depicted the need for adaptive techniques that better cater to threats and vulnerabilities surrounding cloud computing. The observation from the study shows that most works in the literature have no consensus in the design and implementation of effective cloud security schemes, which means that security and privacy implementation in the literature does not balance integrity, accountability, and privacy. Furthermore, cloud models for privacy-preserving are not user-centric, creating no flexibility and control management over security or privacy protocols that maintain users' sensitive data.

Author Contributions: All contributions of this work are as follows: conceptualisation, Y.S.A. and M.H.; methodology, Y.S.A. and M.H.; validation, Y.S.A. and M.H.; analysis, Y.S.A.; investigation, Y.S.A. and M.H.; resources, Y.S.A. and M.H.; writing—original draft preparation, Y.S.A.; writing—review and editing, Y.S.A.; supervision, M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This research was partially supported by Google PhD Fellowship Program. We also like to thank University Mohammed VI Polytechnic in collaboration with Office Chérifien des Phosphates (OCP) Africa for making academic resources available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tari, Z. Security and Privacy in Cloud Computing. *IEEE Cloud Comput.* **2014**, *1*, 54–57.
2. Bentajer, A.; Hedabou, M.; Abouelmehdi, K.; Elfezazi, S. CS-IBE: A data confidentiality system in public cloud storage system. *Procedia Comput. Sci.* **2018**, *141*, 559–564.
3. Fernandez-Gago, C.; Pearson, S.; D'errico, M.; Alnemr, R.; Pulls, T.; de Oliveira, A.S. A4Cloud Workshop: Accountability in the Cloud. In Proceedings of the IFIP International Summer School on Privacy and Identity Management, Edinburgh, UK, 16–21 August 2015; pp. 61–78.
4. Azougaghe, A.; Oualhaj, O.A.; Hedabou, M.; Belkasmi, M.; Kobbane, A. Many-to-one matching game towards secure virtual machines migration in cloud computing. In Proceedings of the 2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS), Marrakesh, Morocco, 17–19 October 2016; pp. 1–7.
5. Mollah, M.B.; Azad, M.A.K.; Vasilakos, A. Security and privacy challenges in mobile cloud computing: Survey and way ahead. *J. Netw. Comput. Appl.* **2017**, *84*, 38–54.
6. Warren, S.D.; Brandeis, L.D. The Right to Privacy Harward Law Review. In *Ethical Issues in the Use of Computers*; Wadsworth Publishing Co.: Belmont, CA, USA, 1890; Volume 4, pp. 193–220.
7. Deng, M. *Privacy Preserving Content Protection (Privacy Behoud Content Protection)*; Faculty of Engineering—Katholieke Universiteit Leuven: Leuven, Belgium, 2010.
8. Priem, B.; Kosta, E.; Kuczerawy, A.; Dumortier, J.; Leenes, R. User-centric privacy-enhancing identity management. In *Digital Privacy*; Springer: New York, NY, USA, 2011; pp. 91–106.
9. Kumar, P.; Sehgal, V.K.; Chauhan, D.S.; Gupta, P.; Diwakar, M. Effective ways of secure, private and trusted cloud computing. *arXiv* **2011**, arXiv:1111.3165.
10. Abdulsalam, Y.S.; Hedabou, M. Decentralized Data Integrity Scheme for Preserving Privacy in Cloud Computing. In Proceedings of the 2021 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Chengdu, China, 18–20 June 2021; pp. 607–612.

11. Sun, X.; Liu, P.; Singhal, A. Toward Cyberresiliency in the Context of Cloud Computing [Resilient Security]. *IEEE Secur. Priv.* **2018**, *16*, 71–75.
12. Chen, D.; Zhao, H. Data security and privacy protection issues in cloud computing. In Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering, Hangzhou, China, 23–25 March 2012; Volume 1, pp. 647–651.
13. Kohnfelder, L.; Garg, P. *The Threats to Our Products*; Microsoft Interface Microsoft Corp.: Albuquerque, NM, USA, 1999; Volume 33.
14. Khan, R.; McLaughlin, K.; Laverty, D.; Sezer, S. STRIDE-based threat modeling for cyber-physical systems. In Proceedings of the 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Turin, Italy, 26–29 September 2017; pp. 1–6.
15. James, J.I.; Shosha, A.F.; Gladyhsev, P. Determining Training Needs for Cloud Infrastructure Investigations Using I-STRIDE. In Proceedings of the International Conference on Digital Forensics and Cyber Crime, Moscow, Russia, 26–27 September 2013; pp. 223–236.
16. Tabrizchi, H.; Rafsanjani, M.K. A survey on security challenges in cloud computing: Issues, threats, and solutions. *J. Supercomput.* **2020**, *76*, 9493–9532.
17. Modi, C.; Patel, D.; Borisaniya, B.; Patel, A.; Rajarajan, M. A survey on security issues and solutions at different layers of Cloud computing. *J. Supercomput.* **2013**, *63*, 561–592.
18. Sgandurra, D.; Lupu, E. Evolution of attacks, threat models, and solutions for virtualized systems. *ACM Comput. Surv.* **2016**, *48*, 1–38.
19. Subramanian, N.; Jeyaraj, A. Recent security challenges in cloud computing. *Comput. Electr. Eng.* **2018**, *71*, 28–42.
20. Subashini, S.; Kavitha, V. A survey on security issues in service delivery models of cloud computing. *J. Netw. Comput. Appl.* **2011**, *34*, 1–11.
21. Zhan, Z.H.; Liu, X.F.; Gong, Y.J.; Zhang, J.; Chung, H.S.H.; Li, Y. Cloud computing resource scheduling and a survey of its evolutionary approaches. *ACM Comput. Surv.* **2015**, *47*, 1–33.
22. Basu, S.; Bardhan, A.; Gupta, K.; Saha, P.; Pal, M.; Bose, M.; Basu, K.; Chaudhury, S.; Sarkar, P. Cloud computing security challenges & solutions-A survey. In Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–10 January 2018; pp. 347–356.
23. Li, R.; Xiao, Y.; Zhang, C.; Song, T.; Hu, C. Cryptographic algorithms for privacy-preserving online applications. *Math. Found. Comput.* **2018**, *1*, 311.
24. Kim, W. Cloud computing: Today and tomorrow. *J. Object Technol.* **2009**, *8*, 65–72.
25. Hedabou, M. Cryptography for Addressing Cloud Computing Security, Privacy, and Trust Issues. In *Computer and Cyber Security*; Auerbach Publications: Boca Raton, FL, USA, 2018; pp. 281–304.
26. Chandramouli, R.; Iorga, M.; Chokhani, S. Cryptographic key management issues and challenges in cloud services. In *Secure Cloud Computing*; Springer: New York, NY, USA, 2014; pp. 1–30.
27. Yang, K.; Jia, X. Data storage auditing service in cloud computing: Challenges, methods and opportunities. *World Wide Web* **2012**, *15*, 409–428.
28. Arunarani, A.; Manjula, D.; Sugumaran, V. Task scheduling techniques in cloud computing: A literature survey. *Future Gener. Comput. Syst.* **2019**, *91*, 407–415.
29. Xiao, Z.; Xiao, Y. Security and privacy in cloud computing. *IEEE Commun. Surv. Tutor.* **2012**, *15*, 843–859.
30. Liu, D. Securing outsourced databases in the cloud. In *Security, Privacy and Trust in Cloud Systems*; Springer: New York, NY, USA, 2014; pp. 259–282.
31. Sookhak, M.; Talebian, H.; Ahmed, E.; Gani, A.; Khan, M.K. A review on remote data auditing in single cloud server: Taxonomy and open issues. *J. Netw. Comput. Appl.* **2014**, *43*, 121–141.
32. Pearson, S.; Benameur, A. Privacy, security and trust issues arising from cloud computing. In Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science, Indianapolis, IN, USA, 30 November–3 December 2010; pp. 693–702.
33. Wu, H.; Zhao, B. Overview of current techniques in remote data auditing. *Appl. Math. Nonlinear Sci.* **2016**, *1*, 140–153.
34. Sookhak, M.; Gani, A.; Talebian, H.; Akhunzada, A.; Khan, S.U.; Buyya, R.; Zomaya, A.Y. Remote data auditing in cloud computing environments: A survey, taxonomy, and open issues. *ACM Comput. Surv.* **2015**, *47*, 1–34.
35. Varghese, B.; Buyya, R. Next generation cloud computing: New trends and research directions. *Future Gener. Comput. Syst.* **2018**, *79*, 849–861.
36. Cook, A.; Robinson, M.; Ferrag, M.A.; Maglaras, L.A.; He, Y.; Jones, K.; Janicke, H. Internet of cloud: Security and privacy issues. In *Cloud Computing for Optimization: Foundations, Applications, and Challenges*; Springer: New York, NY, USA, 2018; pp. 271–301.
37. Tan, Z.; Nagar, U.T.; He, X.; Nanda, P.; Liu, R.P.; Wang, S.; Hu, J. Enhancing big data security with collaborative intrusion detection. *IEEE Cloud Comput.* **2014**, *1*, 27–33. [CrossRef]
38. Wang, C.; Ren, K.; Yu, S.; Urs, K.M.R. Achieving usable and privacy-assured similarity search over outsourced cloud data. In Proceedings of the 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 451–459.
39. Zhou, M.; Zhang, R.; Xie, W.; Qian, W.; Zhou, A. Security and privacy in cloud computing: A survey. In Proceedings of the 2010 Sixth International Conference on Semantics, Knowledge and Grids, Beijing, China, 1–3 November 2010; pp. 105–112.
40. Zou, J. Accountability in Cloud Services. Ph.D. Thesis, Macquarie University, Sydney, Australia, 2016.

41. Goyal, S. Public vs private vs hybrid vs community-cloud computing: A critical review. *Int. J. Comput. Netw. Inf. Secur.* **2014**, *6*, 20. [CrossRef]
42. Hedabou, M.; Abdulsalam, Y.S. Efficient and Secure Implementation of BLS Multisignature Scheme on TPM. In Proceedings of the 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), Arlington, VA, USA, 9–10 November 2020; pp. 1–6.
43. Kamara, S.; Moataz, T. Boolean searchable symmetric encryption with worst-case sub-linear complexity. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, 30 April–4 May 2017; pp. 94–124.
44. Li, P.; Li, J.; Huang, Z.; Li, T.; Gao, C.Z.; Yiu, S.M.; Chen, K. Multi-key privacy-preserving deep learning in cloud computing. *Future Gener. Comput. Syst.* **2017**, *74*, 76–85. [CrossRef]
45. Pearson, S. Taking account of privacy when designing cloud computing services. In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, Vancouver, BC, Canada, 23 May 2009; pp. 44–52.
46. Ko, R.K.; Jagadpramana, P.; Mowbray, M.; Pearson, S.; Kirchberg, M.; Liang, Q.; Lee, B.S. TrustCloud: A framework for accountability and trust in cloud computing. In Proceedings of the 2011 IEEE World Congress on Services, Washington, DC, USA, 4–9 July 2011; pp. 584–588.
47. Patel, A.; Taghavi, M.; Bakhtiyari, K.; JúNior, J.C. An intrusion detection and prevention system in cloud computing: A systematic review. *J. Netw. Comput. Appl.* **2013**, *36*, 25–41. [CrossRef]
48. Li, X.Y.; Zhou, L.T.; Shi, Y.; Guo, Y. A trusted computing environment model in cloud architecture. In Proceedings of the 2010 International Conference on Machine Learning and Cybernetics, Qingdao, China, 11–14 July 2010; Volume 6, pp. 2843–2848.
49. Bertino, E.; Paci, F.; Ferrini, R.; Shang, N. Privacy-preserving digital identity management for cloud computing. *IEEE Data Eng. Bull.* **2009**, *32*, 21–27.
50. Paci, F.; Shang, N.; Steuer Jr, K.; Fernando, R.; Bertino, E. VeryIDX-A privacy preserving digital identity management system for mobile devices. In Proceedings of the 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, Taipei, Taiwan, 18–20 May 2009; pp. 367–368.
51. Wu, H.; Zheng, W.; Chiesa, A.; Popa, R.A.; Stoica, I. DIZK: A Distributed Zero Knowledge Proof System. In Proceedings of the 27th USENIX Security Symposium (USENIX Security 18), Baltimore, MD USA, 12–17 August 2018; pp. 675–692.
52. Hedabou, M. A frobenius map approach for an efficient and secure multiplication on Koblitz curves. *Int. J. Netw. Secur.* **2006**, *3*, 239–243.
53. Wang, C.; Wang, Q.; Ren, K.; Lou, W. Privacy-preserving public auditing for data storage security in cloud computing. In Proceedings of the 2010 Proceedings IEEE Infocom, San Diego, CA, USA, 14–19 March 2010; pp. 1–9.
54. Fiore, D.; Mitrokotsa, A.; Nizzardo, L.; Pagnin, E. Multi-key homomorphic authenticators. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, 4–8 December 2016; pp. 499–530.
55. Garg, N.; Bawa, S. RITS-MHT: Relative indexed and time stamped Merkle hash tree based data auditing protocol for cloud computing. *J. Netw. Comput. Appl.* **2017**, *84*, 1–13. [CrossRef]
56. Ateniese, G.; Di Pietro, R.; Mancini, L.V.; Tsudik, G. Scalable and efficient provable data possession. In Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, Istanbul, Turkey, 22–25 September 2008; pp. 1–10.
57. Erway, C.C.; Küpçü, A.; Papamanthou, C.; Tamassia, R. Dynamic provable data possession. *ACM Trans. Inf. Syst. Secur.* **2015**, *17*, 1–29. [CrossRef]
58. Curtmola, R.; Khan, O.; Burns, R.; Ateniese, G. MR-PDP: Multiple-replica provable data possession. In Proceedings of the 2008 the 28th International Conference on Distributed Computing Systems, Beijing, China, 17–20 June 2008; pp. 411–420.
59. He, D.; Zeadally, S.; Wu, L. Certificateless public auditing scheme for cloud-assisted wireless body area networks. *IEEE Syst. J.* **2015**, *12*, 64–73. [CrossRef]
60. Kang, B.; Wang, J.; Shao, D. Certificateless public auditing with privacy preserving for cloud-assisted wireless body area networks. *Mob. Inf. Syst.* **2017**, *2017*, 2925465. [CrossRef]
61. Garg, N.; Bawa, S.; Kumar, N. An efficient data integrity auditing protocol for cloud computing. *Future Gener. Comput. Syst.* **2020**, *109*, 306–316. [CrossRef]
62. Sookhak, M.; Yu, F.R.; Zomaya, A.Y. Auditing big data storage in cloud computing using divide and conquer tables. *IEEE Trans. Parallel Distrib. Syst.* **2017**, *29*, 999–1012. [CrossRef]
63. Zhang, Y.; Xu, C.; Lin, X.; Shen, X.S. Blockchain-based public integrity verification for cloud storage against procrastinating auditors. *IEEE Trans. Cloud Comput.* **2019**, *9*, 923–937. [CrossRef]
64. Eyal, I.; Gencer, A.E.; Sirer, E.G.; Van Renesse, R. Bitcoin-ng: A scalable blockchain protocol. In Proceedings of the 13th USENIX symposium on networked systems design and implementation (NSDI 16), Santa Clara, CA, USA, 16–18 March 2016; pp. 45–59.
65. McConaghy, T.; Marques, R.; Müller, A.; De Jonghe, D.; McConaghy, T.; McMullen, G.; Henderson, R.; Bellemare, S.; Granzotto, A. *Bigchaindb: A Scalable Blockchain Database*; White Paper; BigChainDB, Ascribe GmbH: Berlin, Germany, 2016.
66. Gaetani, E.; Aniello, L.; Baldoni, R.; Lombardi, F.; Margheri, A.; Sassone, V. Blockchain-based database to ensure data integrity in cloud computing environments. In Proceedings of the 2020 International Conference on Mainstreaming Block Chain Implementation (ICOMBI), Bengaluru, India, 21–22 February 2017.

67. Sari, A. A review of anomaly detection systems in cloud networks and survey of cloud security measures in cloud storage applications. *J. Inf. Secur.* **2015**, *6*, 142. [CrossRef]
68. Farid, D.M.; Rahman, M.Z. Anomaly Network Intrusion Detection Based on Improved Self Adaptive Bayesian Algorithm. *J. Comput.* **2010**, *5*, 23–31. [CrossRef]
69. Feizollah, A.; Anuar, N.B.; Salleh, R.; Amalina, F.; Ma'arof, R.R.; Shamshirband, S. A study of machine learning classifiers for anomaly-based mobile botnet detection. *Malays. J. Comput. Sci.* **2013**, *26*, 251–265.
70. Khorshed, M.T.; Ali, A.S.; Wasimi, S.A. A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing. *Future Gener. Comput. Syst.* **2012**, *28*, 833–851. [CrossRef]
71. Shelke, M.P.K.; Sontakke, M.S.; Gawande, A. Intrusion detection system for cloud computing. *Int. J. Sci. Technol. Res.* **2012**, *1*, 67–71.
72. Wani, A.R.; Rana, Q.; Saxena, U.; Pandey, N. Analysis and Detection of DDoS Attacks on Cloud Computing Environment using Machine Learning Techniques. In Proceedings of the 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 4–6 February 2019; pp. 870–875.
73. Bhamare, D.; Salman, T.; Samaka, M.; Erbad, A.; Jain, R. Feasibility of supervised machine learning for cloud security. In Proceedings of the 2016 International Conference on Information Science and Security (ICISS), Pattaya, Thailand, 19–22 December 2016; pp. 1–5.
74. Rodriguez, R.A. Method of and Apparatus for Combining Artificial Intelligence (AI) Concepts with Event-Driven Security Architectures and Ideas. U.S. Patent 8,583,574, 12 November 2013.
75. Osanaiye, O.; Cai, H.; Choo, K.K.R.; Dehghantanha, A.; Xu, Z.; Dlodlo, M. Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. *EURASIP J. Wirel. Commun. Netw.* **2016**, *2016*, 130. [CrossRef]
76. Gill, S.S.; Buyya, R. SECURE: Self-protection approach in cloud resource management. *IEEE Cloud Comput.* **2018**, *5*, 60–72. [CrossRef]
77. Weyns, D. Software engineering of self-adaptive systems: An organised tour and future challenges. In *Chapter in Handbook of Software Engineering*; Linnaeus University: Kalmar, Sweden, 2017.
78. Acquisti, A.; Gritzalis, S.; Lambrinoudakis, C.; di Vimercati, S. *Digital Privacy: Theory, Technologies, and Practices*; CRC Press: Boca Raton, FL, USA, 2007.
79. Tyagi, N.; Gilad, Y.; Leung, D.; Zaharia, M.; Zeldovich, N. Stadium: A distributed metadata-private messaging system. In Proceedings of the 26th Symposium on Operating Systems Principles. ACM, Shanghai, China, 28–31 October 2017; pp. 423–440.
80. Goldreich, O.; Ostrovsky, R. Software protection and simulation on oblivious RAMs. *J. ACM* **1996**, *43*, 431–473. [CrossRef]
81. Goodrich, M.T.; Mitzenmacher, M.; Ohrimenko, O.; Tamassia, R. Privacy-preserving group data access via stateless oblivious RAM simulation. In Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, Kyoto, Japan, 17–19 January 2012; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2012; pp. 157–167.
82. Stefanov, E.; Van Dijk, M.; Shi, E.; Fletcher, C.; Ren, L.; Yu, X.; Devadas, S. Path ORAM: An extremely simple oblivious RAM protocol. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, Berlin, Germany, 4–8 November 2013; pp. 299–310.
83. Haider, S.K.; van Dijk, M. Flat ORAM: A Simplified Write-Only Oblivious RAM Construction for Secure Processors. *Cryptography* **2019**, *3*, 10. [CrossRef]
84. Di Vimercati, S.D.C.; Foresti, S.; Jajodia, S.; Paraboschi, S.; Samarati, P. A data outsourcing architecture combining cryptography and access control. In Proceedings of the 2007 ACM Workshop on Computer Security Architecture, Fairfax, VR, USA, 2 November 2007; pp. 63–69.
85. Gentry, C.; Fully homomorphic encryption using ideal lattices. In Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, Bethesda, MD, USA, 31 May–2 June 2009; Volume 9, pp. 169–178.
86. Tang, Y.; Lee, P.P.; Lui, J.C.; Perlman, R. FADE: Secure overlay cloud storage with file assured deletion. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Singapore, 7–9 September 2010; pp. 380–397.
87. Fall, D.; Blanc, G.; Okuda, T.; Kadobayashi, Y.; Yamaguchi, S. Toward quantified risk-adaptive access control for multi-tenant cloud computing. In Proceedings of the 6th Joint Workshop on Information Security, Tokyo, Japan, 8–10 November 2011; pp. 1–14.
88. Yu, E.; Cysneiros, L. Designing for privacy and other competing requirements. In Proceedings of the 2nd Symposium on Requirements Engineering for Information Security (SREIS'02), Raleigh, NC, USA, 16–18 October 2002; pp. 15–16.
89. Kobsa, A.; Schreck, J. Privacy through pseudonymity in user-adaptive systems. *ACM Trans. Internet Technol.* **2003**, *3*, 149–183. [CrossRef]
90. Sgaglione, L.; Coppolino, L.; D'Antonio, S.; Mazzeo, G.; Romano, L.; Cotroneo, D.; Scognamiglio, A. Privacy Preserving Intrusion Detection Via Homomorphic Encryption. In Proceedings of the 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Napoli, Italy, 12–14 June 2019; pp. 321–326.
91. Pfitzmann, A.; Hansen, M. A Terminology for Talking about Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management 2010. Available online: http://dud.inf.tu-dresden.de/Anon_Terminology.shtml (accessed on 20 October 2021).
92. Dingledine, R.; Mathewson, N.; Syverson, P. *Tor: The Second-Generation Onion Router*; Technical Report; Naval Research Lab: Washington, DC, USA, 2004.

93. Goldschlag, D.; Reed, M.; Syverson, P. *Onion Routing for Anonymous and Private Internet Connections*; Communication of the ACM; ACM: New York, NY, USA, 1999.
94. Chaum, D. The dining cryptographers problem: Unconditional sender and recipient untraceability. *J. Cryptol.* **1988**, *1*, 65–75. [CrossRef]
95. Bagai, R.; Lu, H.; Li, R.; Tang, B. An accurate system-wide anonymity metric for probabilistic attacks. In Proceedings of the International Symposium on Privacy Enhancing Technologies Symposium, Waterloo, ON, Canada, 27–29 July 2011; pp. 117–133.
96. Diamantopoulou, V.; Kalloniatis, C.; Gritzalis, S.; Mouratidis, H. Supporting privacy by design using privacy process patterns. In Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection, Rome, Italy, 29–31 May 2017; pp. 491–505.
97. Ngai, E.; Ohlman, B.; Tsudik, G.; Uzun, E.; Wählisch, M.; Wood, C.A. Can we make a cake and eat it too? A discussion of ICN security and privacy. *ACM SIGCOMM Comput. Commun. Rev.* **2017**, *47*, 49–54. [CrossRef]
98. Papanikolaou, N.; Pearson, S.; Mont, M.C. Towards natural-language understanding and automated enforcement of privacy rules and regulations in the cloud: Survey and bibliography. In Proceedings of the FTRA International Conference on Secure and Trust Computing, Data Management, and Application, Loutraki, Greece, 28–30 June 2011; pp. 166–173.
99. Chen, T.; Bahsoon, R.; Yao, X. A survey and taxonomy of self-aware and self-adaptive cloud autoscaling systems. *ACM Comput. Surv.* **2018**, *51*, 61. [CrossRef]