Encryption Techniques for Smart Systems Data Security Offloaded to the Cloud.

symmetry

MDPI

Article

Encryption Techniques for Smart Systems Data Security Offloaded to the Cloud

Muhammad Bilal Qureshi [1], Muhammad Shuaib Qureshi [2], Saqib Tahir [3], Aamir Anwar [4], Saddam Hussain [5,*], Mueen Uddin [5] and Chin-Ling Chen [6,7,8,*]

1 Department of Computer Science and IT, University of Lakki Marwat, Lakki Marwat 28420, Pakistan; mbilal@ulm.edu.pk
2 Department of Computer Science, School of Arts & Sciences, University of Central Asia, Naryn 722918, Kyrgyzstan; muhammad.qureshi@ucentralasia.org
3 Department of Computer Science, Shaheed Zulfikar Ali Bhutto Institute of Science & Technology, Islamabad 44000, Pakistan; 1873125@szabist-isb.pk
4 School of Computing & Engineering, University of West London, London W5 5RF, UK; 21452391@student.uwl.ac.uk
5 School of Digital Science, University Brunei Darussalam, Jalan Tungku Link, Gadong BE1410, Brunei; mueen.uddin@ubd.edu.bn
6 School of Information Engineering, Changchun Sci-Tech University, Changchun 130600, China
7 Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan
8 School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China
* Correspondence: saddamicup1993@gmail.com (S.H.); clc@mail.cyut.edu.tw (C.-L.C.)

Abstract: With technological advancement, cloud computing paradigms are gaining massive popularity in the ever-changing technological advancement. The main objective of the cloud computing system is to provide on-demand storage and computing resources to the users on the pay-per-use policy. It allows small businesses to use top-notch infrastructure at low expense. However, due to the cloud resource sharing property, data privacy and security are significant concerns and barriers for smart systems to constantly transfer generated data to the cloud computing resources, which a third-party provider manages. Many encryption techniques have been proposed to cope with data security issues. In this paper, different existing data protection and encryption techniques based on common parameters have been critically analyzed and their workflows are graphically presented. This survey aims to collect existing data encryption techniques widely presented in the literature for smart system data security offloaded to the cloud computing systems under a single umbrella.

Keywords: cloud computing; smart systems; data protection; encryption techniques; access control; symmetric and asymmetric encryption

Citation: Qureshi, M.B.; Qureshi, M.S.; Tahir, S.; Anwar, A.; Hussain, S.; Uddin, M.; Chen, C.-L. Encryption Techniques for Smart Systems Data Security Offloaded to the Cloud. *Symmetry* **2022**, *14*, 695. https://doi.org/10.3390/sym14040695

Academic Editor: Tomohiro Inagaki

Received: 19 February 2022
Accepted: 22 March 2022
Published: 28 March 2022

1. Introduction

The name cloud computing was inspired by the cloud symbol that is usually used to represent the Internet in flowcharts and diagrams. Cloud computing provides a platform for hosting anything that needs resources over the Internet. The platform services provided by cloud computing are broadly divided into three categories: Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS), and Software-as-a-Service (SaaS). Cloud computing provides shared computing resources rather than having dedicated servers. Applications, storage, and other host services are accessed via a web portal. The payment methodology can be on an as-needed or pay-as-you-go business model. Cloud computing enables companies to consume shared resources as utilities rather than having to build and maintain computing infrastructures on their own. Different types of resources can be shared, such as a virtual machine (VM/container), storage, or an application. The cloud computing paradigms are attractive because the organizations can avoid the installation

and maintenance cost of the IT infrastructure by hiring the resources by using a simple pay and use policy for what they want to use and when [1]. Furthermore, cloud computing providers can benefit from significant savings by delivering the same services to a wide range of customers, especially smart system stakeholders. The general architecture of secure data offloading from smart system devices to the cloud computing system is depicted in Figure 1.

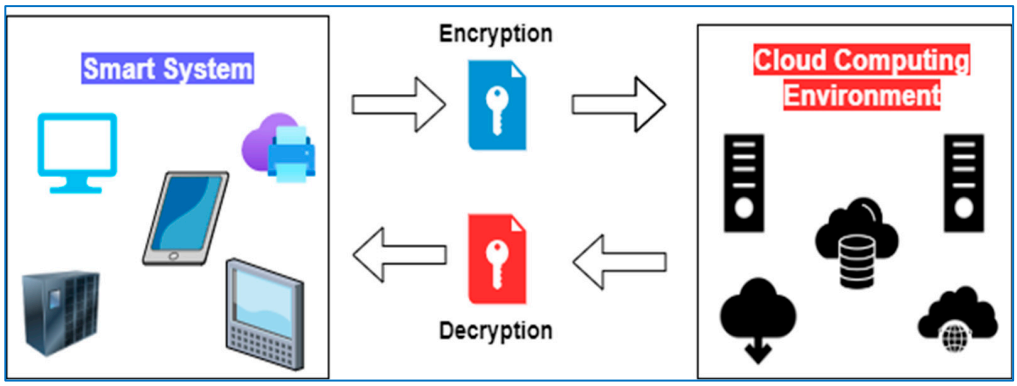

Figure 1. The smart system's general model of secure data offloading to the cloud computing environment.

Some of the most significant barriers and obstacles in the rapid growth and adoption of cloud computing across the industry are data security and privacy issues. In this digital day and age, the amount of data generated by smart system devices is increasing every day, which results in the demand for more data storage and faster processing. It is a top aim for every organization to reduce data storage and processing costs somehow while maintaining the same level of analysis of data and information. However, the confidentiality of the data is paramount because the data is traveling between the smart system and cloud computing resources. Companies need to have a trust level with the model to adopt cloud computing. In order to achieve a higher trust level and ensure their data is being transferred securely, encryption algorithms are a solution to safeguard confidential data. Researchers have proposed many algorithms to address trust issues and attain a high level of cloud data security. The generalized model of cloud data security is provided in Figure 2.

Cloud computing utilizes shared infrastructure resources to perform data analysis and provide a higher level of service (redundancy) without maintenance and management overhead [2]. Any data being transferred must be safe from malicious interceptions and data privacy breaches. There are ways to manage this transfer from an architectural standpoint, but all such methods leave data in plain text that any attacker can understand. The encryption process is critical in the cloud computing domain as data is transferred over the Internet to the cloud computing platform from local servers and, in many cases, between cloud computing resources (VM1 to other VMn). Since all this transfer takes place over the Internet, it is a significant security and data privacy risk if data is sent as plain text. There are many ways to intercept this data over the networks that demand more to be done to safeguard data. Encryption resolves this problem to a great extent by ensuring that the data is not decipherable during transfer.

Encryption converts data into another form, which can only be deciphered by users with the respective keys or other access mechanisms [2]. Encrypted data is commonly called ciphertext, while decrypted data is plain text. Encryption can be broadly classified into symmetric encryption, public-key encryption, and asymmetric encryption, also known as private-key encryption [3]. When encryption and decryption are performed using the same key for both functions, it is called symmetric key encryption, whereas, for asymmetric encryption, a public-private key pair is generated where public key encrypts and private key decrypts the data. The general classification of encryption algorithms is sketched in Figure 3. One of the benefits of asymmetric encryption is that the distribution of keys can

be controlled better and more securely. For instance, if someone gets access to the public key, it is useless and cannot decrypt the data. It is imperative to define a proper method to distribute keys to ensure that they are not in the hand of rogue individuals/devices. The general architecture and decrypt procedure of symmetric and asymmetric key algorithms are drawn in Figures 4 and 5.

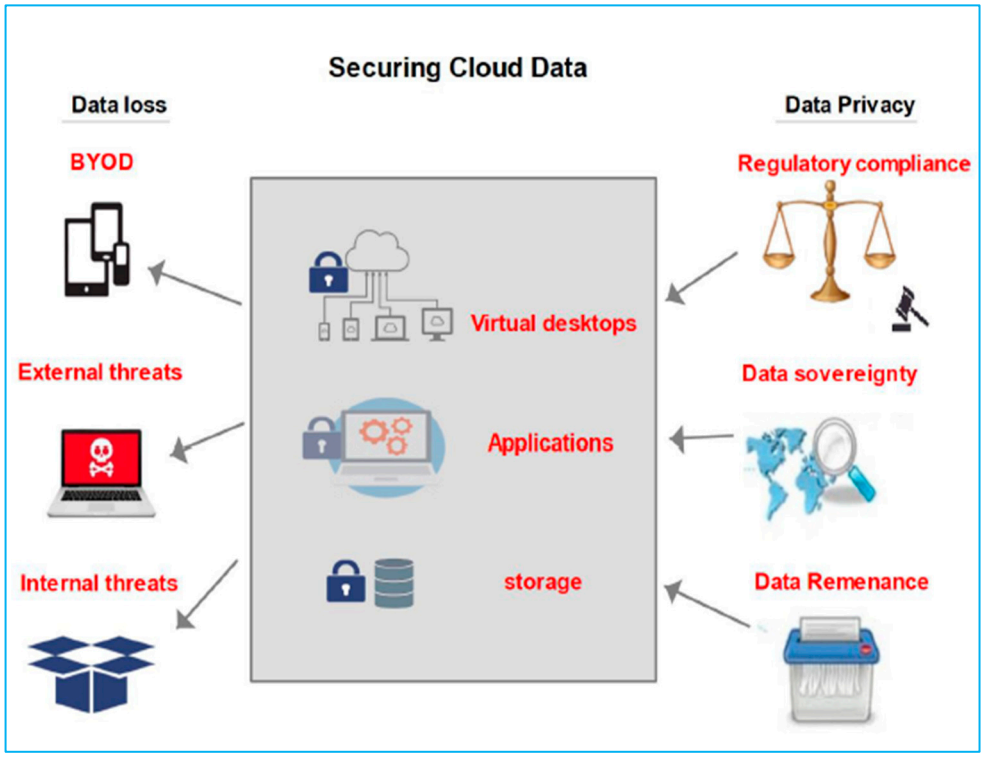

Figure 2. Cloud data security model.

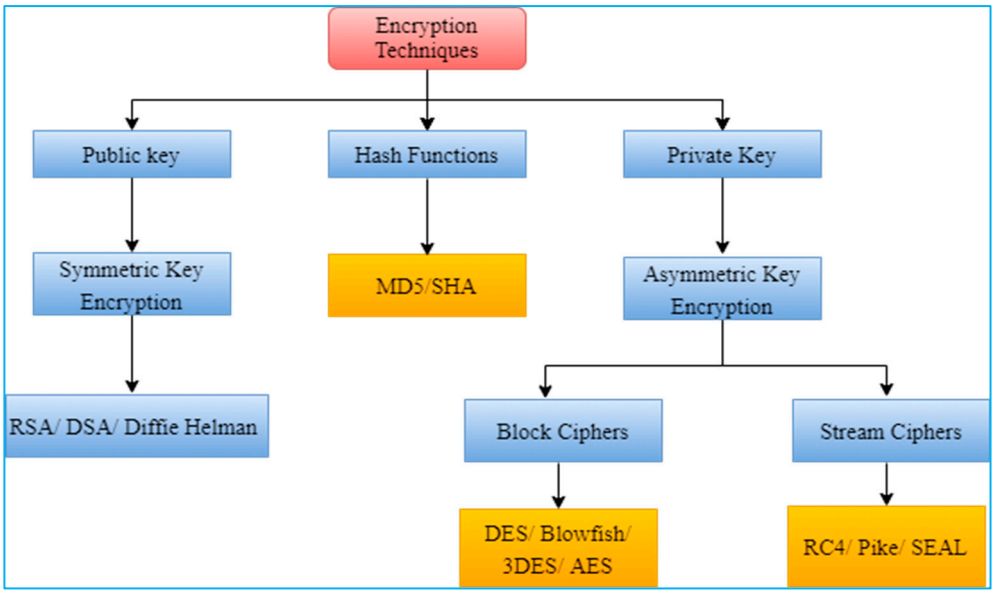

Figure 3. Taxonomy of encryption algorithms.

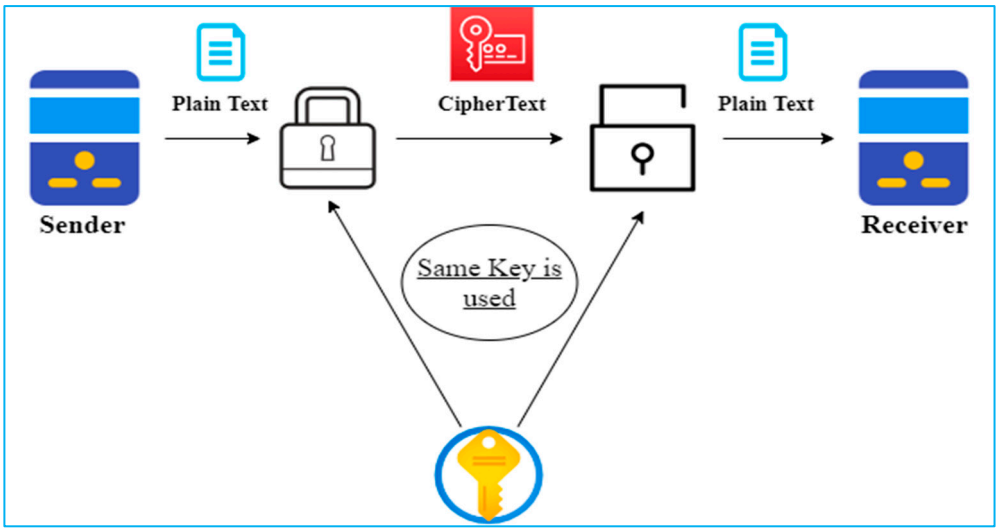

Figure 4. The general architecture of symmetric key encryption.

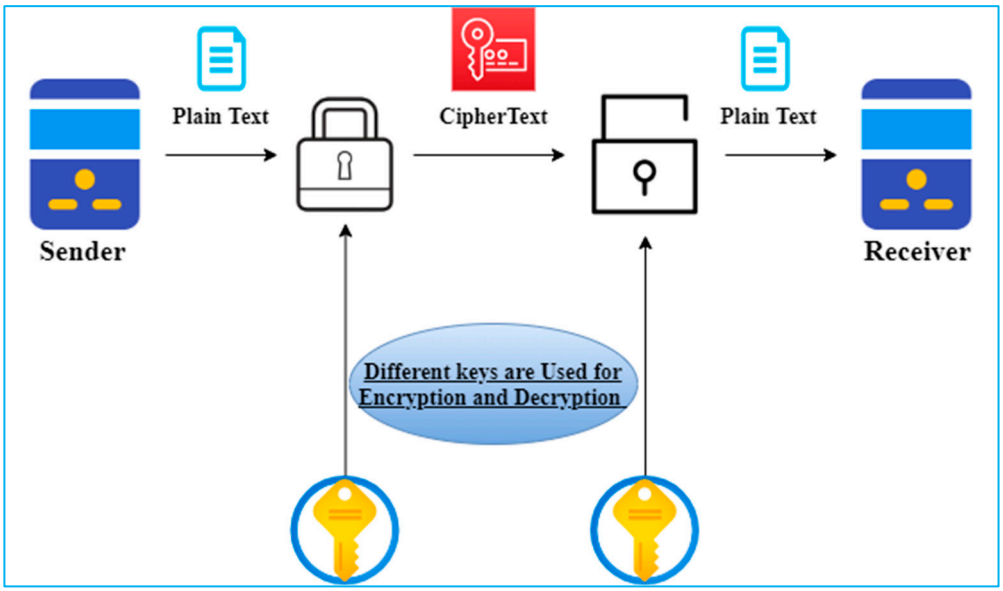

Figure 5. General architecture of asymmetric key encryption.

Pros and Cons of Encryption/Decryption

- Data encryption is the way of permitting data by keeping it distinct from the device on which it is stored. The administrators can store and send data via insecure channels.
- Data encryption ensures the safety of sensitive information and intellectual property.
- The data is secure regardless of its transmission because encryption is built into the data.
- Many organizations follow strict rules and confidentiality guidelines. The encryption paves the way because the data can only be viewed by the recipient who possesses the key to decrypt it.
- Data encryption can be fairly costly because the systems that keep it up to date must have the capacity and improvements.

The main contributions of this survey are summarized as follows:

- To search and investigate the latest number of related published research work on "cloud computing data security" and "encryption algorithms for cloud computing" in Google Scholar and other known digital libraries.

- To review the general and key distribution architecture of encryption techniques by considering the cloud computing infrastructure and platform requirements.
- To identify the security gaps, encryption key size, and encryption time.
- To address the strengths and weaknesses of the studied encryption methodologies, platforms, and applications used for smart systems data offloading to the cloud computing systems.
- Studying different encryption techniques for smart systems data security allows us to decide better which technique to use depending on the requirements. Complete knowledge of each technique is vital in execution time, complexity, and maintainability.

The rest of the paper is organized as follows: Section 2 discusses existing data security techniques for smart systems in the cloud computing environment. Section 3 gives a comparative analysis of the existing data encryption techniques used for smart systems data security. Section 4 presents smart systems data security challenges in cloud computing infrastructure. The conclusion and future work are outlined in Section 5.

2. Literature Review

The focus on cloud security can be observed from the publishing trends of the research community. To investigate this trend, we analyzed the search occurrences of encryption techniques and cloud security in Google Scholar, while many published papers with the same titles in four major digital libraries are Science Direct, ACM, Web of Science, and IEEE Xplore. Figure 6 shows search occurrences of the keywords mentioned earlier for the last ten (10) years in Google Scholar. The graph shows that cloud security was the primary focus of the research community until 2015 but gradually decreased due to the emerging trend of similar technologies, such as edge and fog computing. Compared to cloud security, the graph for the cloud encryption techniques remained consistent. The scholarly searches for cloud encryption techniques are increasing year by year. It indicates that encryption techniques are the fastest-growing research area in academia.

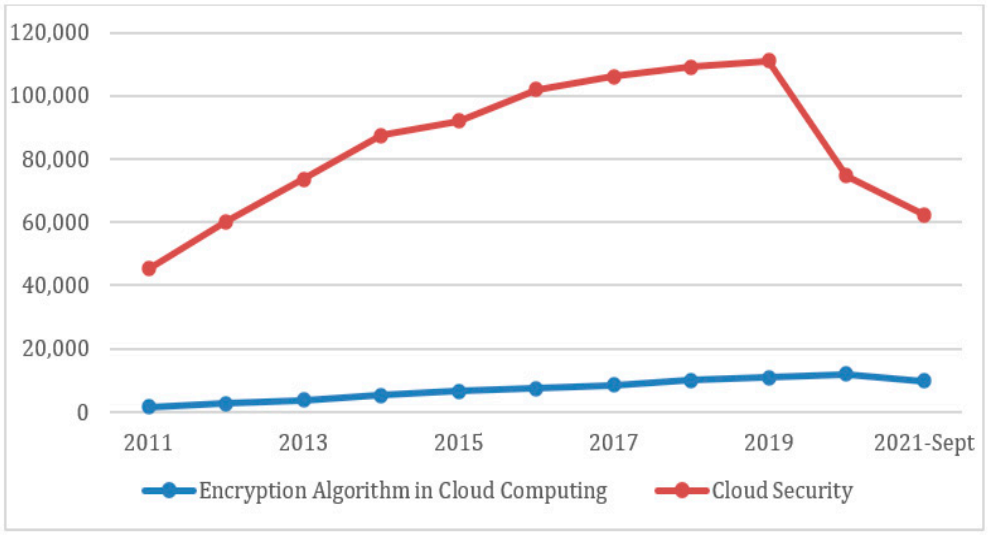

Figure 6. Search occurrences of encryption and cloud data security techniques in Google Scholar.

The total number of papers published on cloud encryption techniques and cloud security in different digital libraries, such as Science Direct, ACM, and Springer Link, is shown in Figure 7.

The following encryption techniques are detailed for data security in cloud computing, which can help in strategic decision-making to improve data offloading to the cloud computing resources.

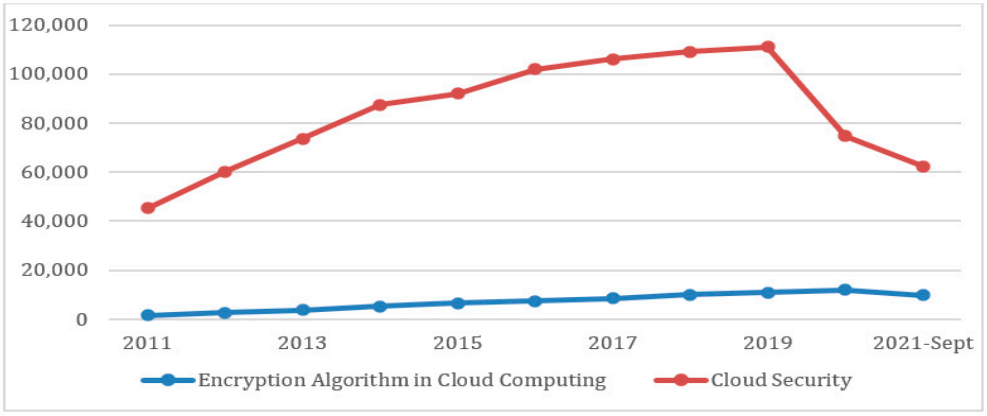

Figure 7. Many publications have "encryption techniques" and "cloud data security" in ACM, Springer Link, and Science Direct.

2.1. Blowfish with Compressed File

Cloud platforms allow the users to use the shared resources without investing much or taking care of server maintenance. Data security is the primary concern in adopting cloud computing technology. Grover et al. [4] suggested Blowfish with a compressed file mechanism for securing data before storing it on the cloud storage resources using encryption techniques and reducing the storage space. The file is first compressed and then encrypted using the Blowfish algorithm, as shown pictorially in Figure 8.

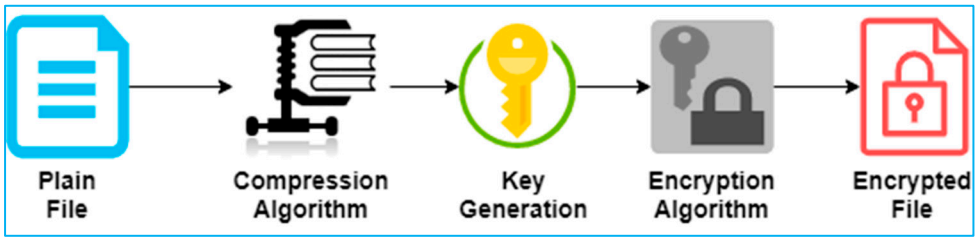

Figure 8. Blowfish encryption process [4].

The proposed methodology resolves the issue of reducing encryption time and space by compressing the data file. The time efficiency of the proposed approach is tested by using three different file sizes. The results are then compared by calculating encryption time for non-compressed and compressed files. Using this approach, if the file is compressed before applying encryption strategy, encryption time and the space required to save this file can be decreased. However, for larger files sizes, its performance is not discussed. As the number of users increases, key management will be a challenging issue.

The Blowfish encryption approach is applied for data security on the client-side, which saves time consumed on the server-side instead. The weakness of this approach is that as the file size increases, more time is required for compression and encryption, and hence, performance is compromised.

2.2. RSA with AES

Khanezaei et al. [5] used slightly different encryption techniques; they have used the combination of Rivest Shamir Adelman (RSA) and Advanced Encryption Standard (AES) for encryption. With the double encryption method, more secure data is shared among the users. RSA is used to increase the complexity of the encrypted text, and AES is used for the fast retrieval of data. The use of RSA for transferring files is secured due to generating asymmetric keys, which is a time-consuming process.

The cloud service generates a public key (PB), a private key (PK), file ID, and a big random number (RB). At first, the user requests the PB from the cloud. The cloud system

provides the PB and file ID to the user. The user then sends the file encrypted using RSA to the cloud. When the user requests a specific file from the cloud service, it sends a request to the server with the public key. The cloud service finds the requested file in the Cloud Storage System (CSS). This file is then encrypted using the AES. The RB, which is the secret key of a symmetric algorithm, is encrypted using the public key. The CSS then sends the RB and requested file to the user. The symmetric algorithm is used because of the key distribution issue, but how to manage these keys is not discussed.

The strength of this approach is that it uses a hybrid approach by using RSA and AES encryption methods for providing data security. The use of RSA increases the difficulty level to hack the data, whereas AES reduces the time required to transfer files between the user and cloud data storage. The major drawback of the proposed approach is that if the file size increases, the number of keys also generated increases, giving rise to the key management issue. Another weakness is that encryption and decryption time is also an overhead for large file sizes.

2.3. Advanced Encryption Standard

With the advancement in cloud computing technology, cloud service providers must resolve data confidentiality and security issues. Sachdev et al. [6] used the AES data security approach, which is used to encrypt data before sending it to the cloud.

The AES works better in software and hardware environments, both in 8-bit and 64-bit platforms. AES requires less memory, which makes it suitable for environments that have less space. AES keys are easy to set up and support any block, and the size of the keys of multiple of 32 must be greater than 128. Data are sent to the cloud service provider after it is encrypted at the user end using the AES algorithm. The user manages the data and key itself, which ensures data integrity. AES provides less memory consumption and computational time than other encryption techniques. In order to maintain the keys, this approach suggests installing a separate physical server at the user end, which increases the hardware cost.

The strength of the AES approach is that data encryption and decryption is performed by the user rather than the cloud service provider.

2.4. Fully Homomorphic Encryption

Zhao et al. [7] have used the Fully Homomorphic Encryption technique, allowing users to do computations on encrypted data without decrypting it. Hence, in this technique, the user can work on ciphertext without exposing its original data to the cloud, which provides extra security to the data.

The fully homomorphic encryption technique results are analyzed by simply using addition and multiplication. Multiplication and addition are applied to the ciphertext, and their results are compared with the plain text. The computation on ciphertext is more complex than plain text. Therefore, such complex computation will be very time-consuming for a small set of data. The user first logs into the cloud select storage section based on the data security level. If the private section is selected, then the AES algorithm is used for encryption, whereas the Blowfish encryption technique is used for the public section. Two data encryption models are offered for hybrid data storage, which provides security in the private or public section. Simplified Advanced Encryption Standard (S-AES), International Data Encryption Algorithm (IDEA), or Blowfish encryption technique can be selected if low-level security is required. First, Blowfish encryption is applied for high-level security, and then IDEA is used. After a file is encrypted, the Secure Hashing Algorithm (SHA-1) is used to generate the integrity code. This code is added to the start of the encrypted file, and again, SHA-1 is applied to generate an alphanumeric token of 16 digits. For retrieving the file, the user is first authenticated, after which the SHA-1 token is compared with the one provided at the time of retrieval to maintain the data integrity.

The strength of the Fully Homomorphic Encryption approach is that it provides security in public, private, and hybrid storage sections by using different encryption algorithms

with integrity verification schemes. The user selects the storage section of the cloud according to its security requirements. The private section provides the highest security by using the AES algorithm. In the public section, security provision is limited. This section works best if a user wants fast computation and less encryption and decryption time.

2.5. Novel Encryption Techniques

Kaur et al. [8] provided encryption according to the storage section selected by the user. Blowfish encryption is used for the private, AES, and public sections, as shown in Figure 9a,b. In the hybrid section, the user can select from two tiers which encryption method is the best according to its need. If less security is required, then any S-AES, IDEA, or Blowfish methods are selected, but for high data privacy, Blowfish is first used, and then IDEA is implemented on the already encrypted data. This methodology provides data security by working in two phases: the first phase is storing data securely on the cloud and the second phase deals with the data retrieval from the cloud using double authentication and integrity verification. Hence, this technique provides data security at both data storing and retrieval phases, as shown in Figure 9c.

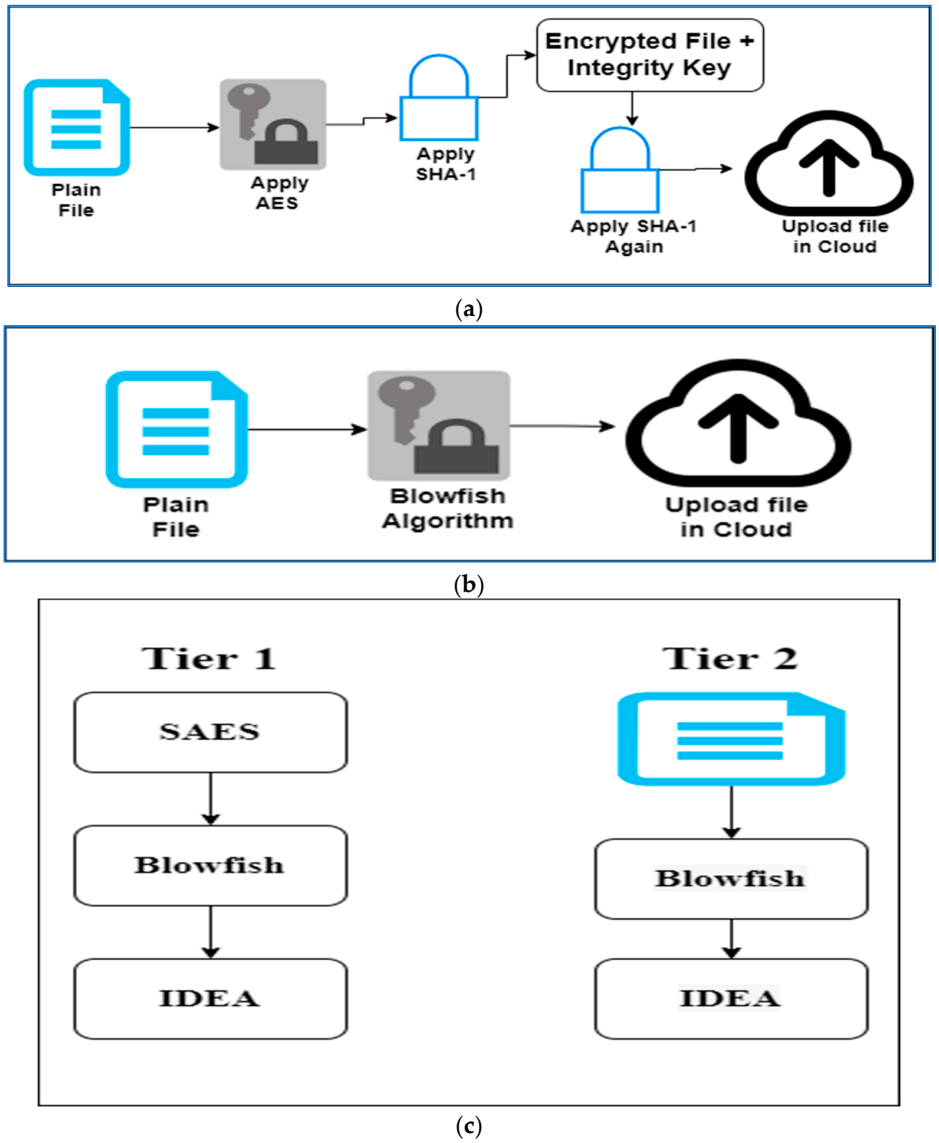

Figure 9. (**a**) Private section. (**b**) Public section. (**c**) Hybrid Section. The encryption process of private, public, and hybrid sections of Novel Encryption Techniques [8].

The Novel Encryption technique provides confidentiality by storing data on clouds in cipher format using different encryption techniques; each technique serves different purposes. The use of SHA-1 ensures data integrity by generating integrity checking code when uploading an encrypted file to the cloud, which is verified at the time of downloading. This approach restricts the unauthorized user from accessing data by implementing a double authentication method, and data tempering is also secured by using an integrity verification scheme.

2.6. Hybrid Techniques with Secure Endpoint

Rani et al. [9] used public and private key encryption techniques to hide the users' sensitive data and ciphertext retrieval. The hybrid technique provides security at the time of login to the cloud system. In this approach, the username is provided in plain text and the password in encrypted form using different encryption algorithms. The username and password are then compared with the already stored login credentials to check the authenticity of the user. This method can speed up the process of getting the root cause of data tempering because if any user is tempered with the data, it will be identified by the username and password stored in the cloud. The username is in plain text, and the password is encrypted by using Caesar cipher. Afterward, the encrypted result is again encrypted by using the RSA algorithm. Then this result is encrypted using monoalphabetic substitution. The authors suggest that the passwords should be encrypted three times to provide better security, but it increases the encryption time three times in return, as shown in Figure 10.

The strength of the hybrid technique is that it is a combination of Ceaser cipher, RSA, and monoalphabetic substitution to encrypt the message. The main drawback of the proposed technique is that it uses a combination of encryption algorithms for encrypting passwords, which increases the encryption time. Computational complexity is also increased due to the application of multiple encryption techniques as compared to the previously encrypted result.

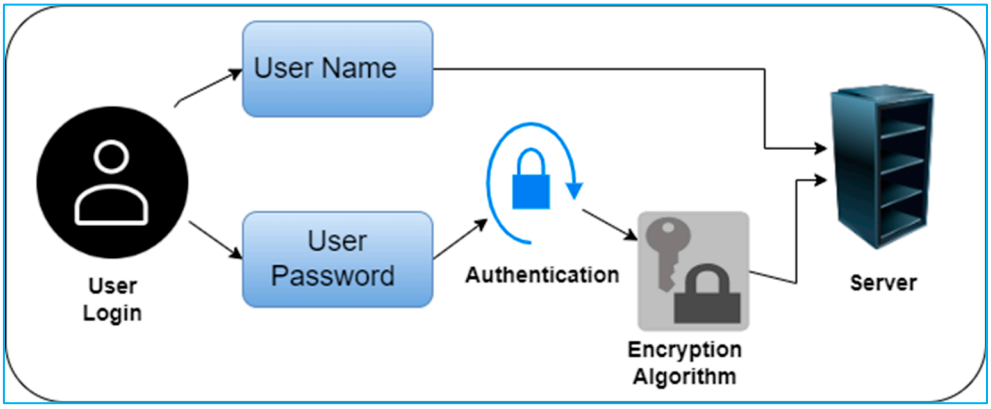

Figure 10. Flow architecture of Hybrid Techniques with Secure Endpoint [9].

2.7. Cluster Load Balancing over AES

Gai et al. [10] proposed the Cluster Load Balancing method. In this model, some essential security services, including authentication, confidentiality, and integrity, are provided using AES. The AES encryption technique is used for data privacy purposes. The proposed methodology is Dynamic Data Encryption Strategy (D2ES), in which the data is encrypted based on the privacy weight attached to it.

The privacy weight determines only the data with the highest privacy value to be the encrypted value, reducing encryption time by selecting highly sensitive data. In this procedure, first, the amount of data for the data package type is input, and then the execution time is calculated for encrypted and unencrypted data. Afterward, the Privacy Weigh Value (PWV) for each data type is calculated, determining the encrypted data's

privacy importance. The D2ES works mainly in three phases. The first phase calculates the weight of the data for privacy and sorts them in descending order. In the second phase, the number of data packets of different data packages is selected for encryption. The third phase encrypts the highest privacy-weighted data, as shown in Figure 11.

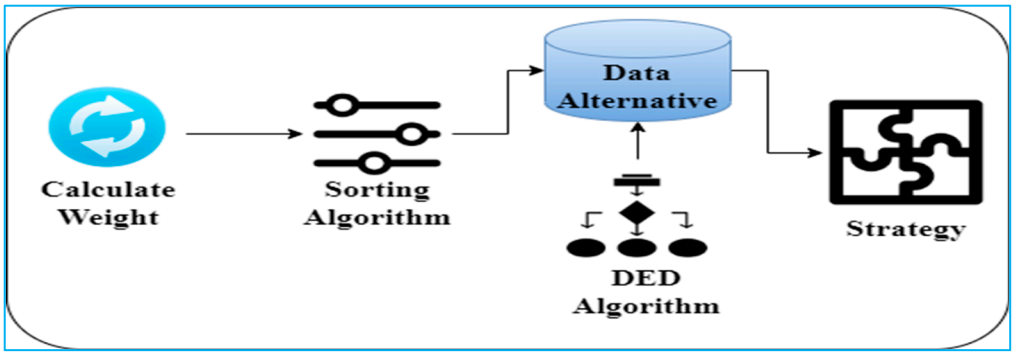

Figure 11. Working phases of the Dynamic Data Encryption Strategy.

The best feature of Cluster Load Balancing over the AES technique is that it uses D2ES for data privacy where the data packages are classified according to the privacy level within a certain time constraint, and then the data is processed for encryption or decryption based on privacy level. The weakness of the technique mentioned above is that it does not describe how to transfer data types with the same priority.

2.8. Three-Layer Security Structure

The Three-layer Security Structure technique is discussed in [11], implemented by the Amazon EC2 cloud service provider. Different encryption algorithms are used to provide security at different levels. The Blowfish or DES techniques are preferred whenever a low encryption time performance of the security algorithm is required rather than higher security of data. However, when high data security is required, AES is preferred.

The working model is three-layered centric, where each layer is responsible for data security. The model provides security to the data layer by layer. Amazon EC2 is used as a case study. The most appropriate technique is selected based on encryption time and speed based on the cloud infrastructure. AES is the most suitable for Amazon EC2 because of its high security and less encryption time. The Three-layer Security Structure performs the best when performance and speed are required. The prominent feature of this approach is that it provides a data security model based on the host cloud architecture.

2.9. RSA with MD5 Hash Algorithm

In [12], the authors provided a data security environment using the Java platform. The best characteristic of the proposed technique is that if the cloud admin wants to read or update, they will get permission from the client environment. Similarly, the user asks for permission from the cloud to update and read. The data is encrypted using MD5 while it is uploaded using RSA. The user is identified for updating in a cloud environment by providing a secure key, which is sent with a tag during data upload. If the cloud detects the changed tag, it will notice, and data access is prevented.

The working procedure of the proposed technique addresses the trust issues when operating in the cloud by transferring the data via APIs, which are always encrypted, alleviating the concerns of insecure data transfer. Data loss concern is also addressed since the cloud admin cannot modify data. The main feature of the proposed technique is the implementation of double security measures with RSA and message digest, which ensures data integrity and security.

The cloud admin can decrypt the information but cannot change it unless the MD5 hash matches. The administrative overhead of the proposed technique is managing public-

private key pairs to encrypt data at cloud user and decrypt at cloud admin. In the case of any data breach, a new private-public key pair will need to be generated every time.

2.10. RSA with Digital Signature

The authors in [13] handle cloud data security and privacy by using a digital signature method that allows users to believe that an invalid user does not create the data.

The proposed methodology uses digital signatures to provide data security. The authenticity of the data is checked when a valid user sends some data from cloud to another user. The proposed method chomps down the document into a few lines by using a hashing algorithm called the message digest. This message digest is then encrypted by using the private key. The signature is decrypted into the message digest with the sender's public key and its private key, as shown in Figure 12.

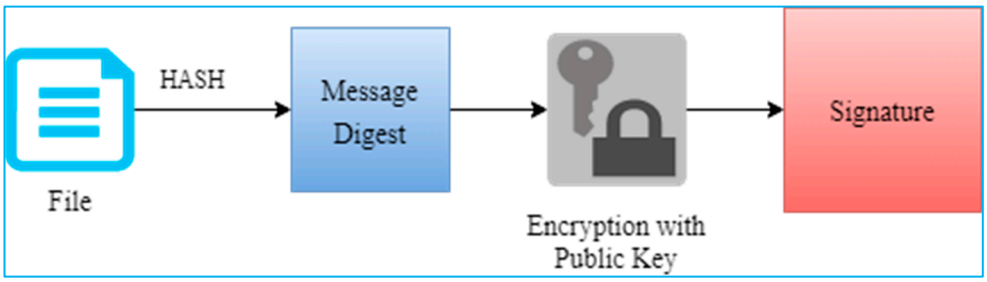

Figure 12. Internal working steps of RSA with the Digital Signature technique.

The digital signature is used to authenticate the message. If the signature is valid, the receiver will know that the valid user sends the message, and it is not tempered; by using an asymmetric encryption algorithm and then again encrypted using the public key.

The proposed hybrid methodology uses a double-encryption process before uploading the data to the cloud. It works so that the data file is first encrypted with symmetric Data Encryption Key (DEK), and then the DEK is encrypted using the CP-ABE algorithm. The author used hybrid approaches of encryption algorithms to achieve maximum security. CP-ABE is used with bi-linear pairing computation to maintain the system's efficiency and size of the ciphertext. It also works on untrusted cloud servers because of attribute-based access requests. The shortcoming of this approach is that it only works on limited file formats, such as text, pdf, and word. Furthermore, the user has to remember the 16-bit security key.

2.11. Security-Aware Efficient Distributed Storage Technique (SA-EDS)

The SA-EDS model [14] consists of two components; the first component is the verified Deterministic Process (DP), and the second is the Data Distributed Storage Process (D2SP). The DP verifies the high-security level of the data, while D2SP is used to protect data from unethical activities. Before sending data on the cloud, the original data is split into two parts. Both parts are encrypted with Alternative Data Distribution (AD2), Secure Efficient Data Distribution (SED2), and Efficient Data Conflation (EDCon) algorithms. The proposed model operates in two steps: splitting and retrieving data packets. In the splitting step, the plain text is divided into sub-parts according to the information security level. After encryption, the sensitive part is stored on server A, and the other is stored on server B. When the user needs to retrieve their data from cloud servers, as shown in Figure 13, the user gets the ciphertext and XOR with the corresponding key to transform it into the original text.

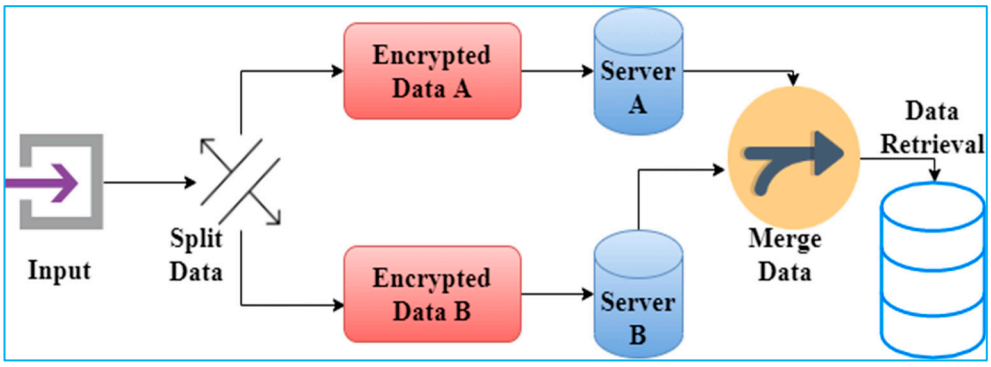

Figure 13. Flow steps of SA-EDS [14].

The strength of this model is splitting data into two parts and sending it into different cloud servers. By using different servers, the security complexity is increased. The model's weakness is increasing the retrieval time from the servers and then converting it into original form.

2.12. Centralized Key Management System (CKMS) with RSA Encryption Algorithm

The encryption/decryption key management deals with the functionalities to generate and store the key and its distribution among two parties. In [15], K. V. Pradeep et al. used Centralized Key Management System (CKMS) with an RSA encryption algorithm. In this approach, the modified Diffie–Hellman distribution schemes are used to distribute keys, which are further used to encrypt the data. When uploading a file to the cloud server, the CKMS is used to encrypt the file using the private key according to the associated file owner. Then encryption on the file is carried out. All CKMS is performed by using Certificate Authority (CA). The CA verifies the server name, certificate directory, registration authority, and key generator. The public key is divided into two parts by using CKMS; one part is sent to the user, and the rest of the part is kept by the CKMS, as shown in Figure 14. The whole key is used for decrypting the file.

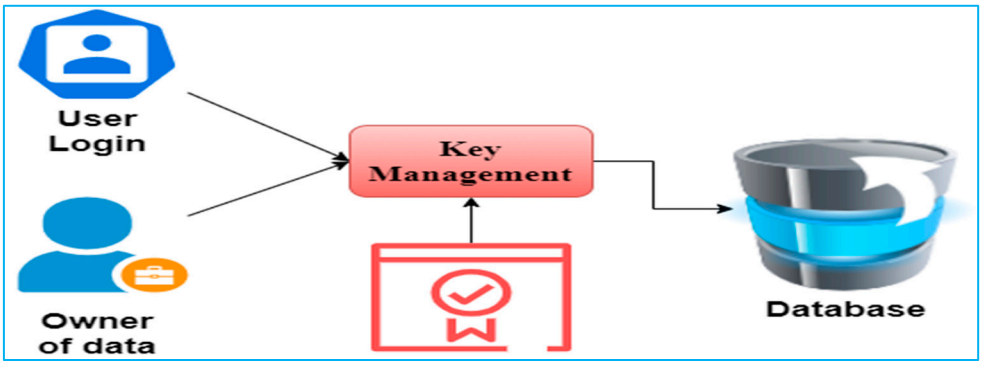

Figure 14. CKMS working model [15].

The strength of this method is that it provides secure data storage on a cloud server. It is hard to access due to the registration process.

2.13. Triple Level Encryption

To mitigate the security problems in cloud computing, Chandrika et al. proposed a method in [16] by using triple-level encryption. They encrypt a file with a different encryption algorithm to achieve high security at each level. The DES algorithm is used during file uploading to achieve first-level encryption in the first level. Then the AES technique is used to achieve the second level, and in the third level, RSA is used for

encryption. After that, they store the ciphertext in a database, as shown in Figure 15. At the time of decryption, RSA is used to decrypt the ciphertext in the first level and then AES to decrypt the next level. Then decryption is performed with DES in the last level to convert it into plain text.

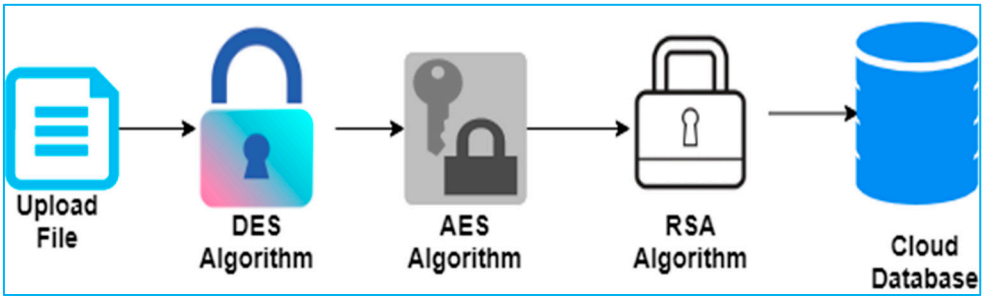

Figure 15. Flow Steps in [16].

The main flaw in this approach is its increasing time complexity and file access time.

2.14. Encryption with Secure HTTPS Connection

The architectural layout of cloud computing is broken into the application, storage, and connectivity segments. Each segment offers different products and services for businesses around the world. In this approach, both the application and architecture levels security are ensured. The file is submitted to the cipher cloud through an encrypted connection when the user clicks the upload button. The encrypted connection is secured by using HTTPS. The double encryption is performed by using a symmetric key algorithm chosen by the user, as shown in Figure 16 [17].

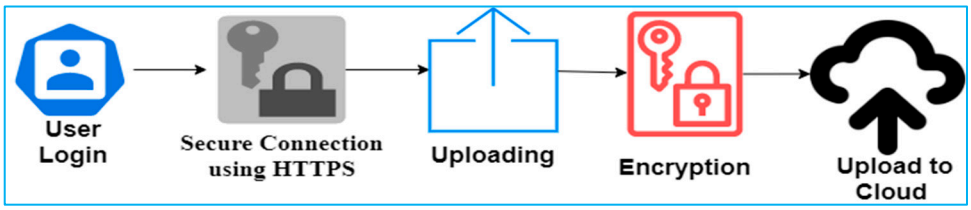

Figure 16. Flow steps of encrypting data with secure HTTPS connection [17].

The flavor of using this approach is that the user chooses the encryption algorithm by itself according to their requirements.

2.15. AWS Logins with Homomorphic Encryption

Poteya et al. proposed a scheme in [18] to perform encryption on data by using a homomorphic algorithm. This version is used with AWS public cloud (DynamoDB) to secure users' data. Based on the operation requirements, the user logs in using their credentials. The AWS cloud database offers services of storing and accessing user data through the login module after verifying the user details by the key selection component stored in the database. The data is stored in an encrypted format through the encryption component and retrieved by using the decryption component. The AWS computation component performs operations on the user data as per the user requirements or the query fired. The significant advantage of the discussed scheme is that it provides confidentiality to the data because, in every phase, no data is exposed in plain text.

2.16. Double Encryption Model

The double encryption model encrypts the data two times to ensure more data security before uploading to the cloud. The algorithm in [19] encrypts plain text with the AES

algorithm, and then the AES key is encrypted using the RSA-1024 algorithm. The working procedure of the double encryption scheme is portrayed in Figure 17.

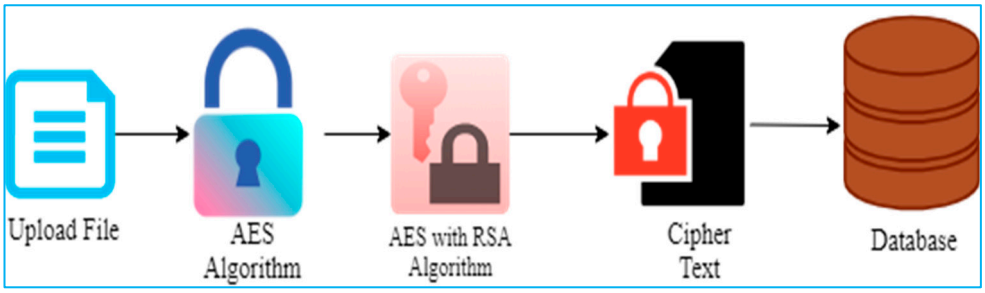

Figure 17. Double encryption model.

The strength of this method is that AES decreases the time complexity of file sharing, and RSA ensures high security. The weakness of this method is key management and encryption/decryption overhead for the large files.

2.17. Hybrid Encryption with MD5 Hash Function

Salma D. et al. in [20], have introduced the hybrid technique, which uses a combination of symmetric and asymmetric algorithms and hashing key using the MD5 hashing function. In the encryption phase, the whole plain text is divided into n blocks, and each block is further divided into two parts; one part is encrypted with AES and the other with the Blowfish algorithm. The resultant text is a single block. The key used in the encryption process is hashed using the MD5 hashing function, as depicted in Figure 18. In the decryption phase, all ciphertext is divided into n blocks, like in the encryption phase, and then each block is further divided into two parts. Each part decrypts the ciphertext by using their respective algorithm used in the encryption phase with hashed key. The efficiency of the hybrid technique can be tested according to the size of the ciphertext, time of encryption/decryption, and throughput.

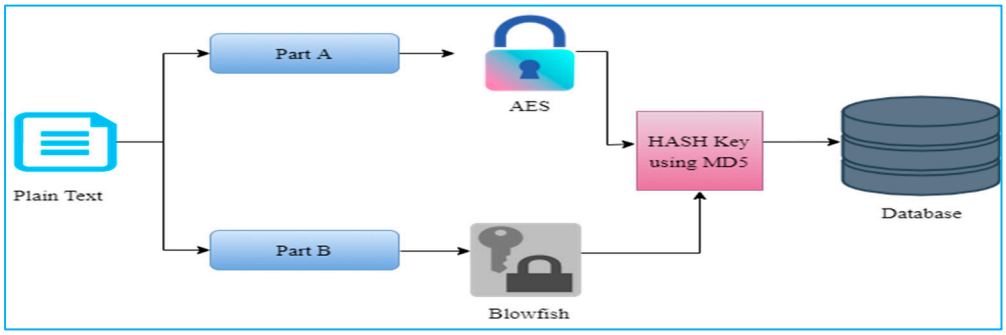

Figure 18. Hybrid encryption process with MD5 hash function [20].

2.18. Homomorphic Encryption with Multi-party Computation

The authors of [21] proposed an encryption technique that integrates multi-party computation and homomorphic encryption techniques to maintain the confidentially and integrity of the data in the cloud environment. The proposed method works in three steps: Homomorphic Encryption (HE), Multi-party Computation, and Key Generation, Encryption, and Decryption. The prominent feature of this method is combining Homomorphic encryption and Multi-Party Computation (HE+MPC), due to which the confidentiality and integrity of the data are maintained with less overhead.

2.19. RSA with Partitioned File

The RSA with the partitioned file technique split the plain text into the same size blocks. Each block is encrypted individually with the RSA algorithm. Then each encrypted block is separately distributed in the cloud server, as shown in Figure 19.

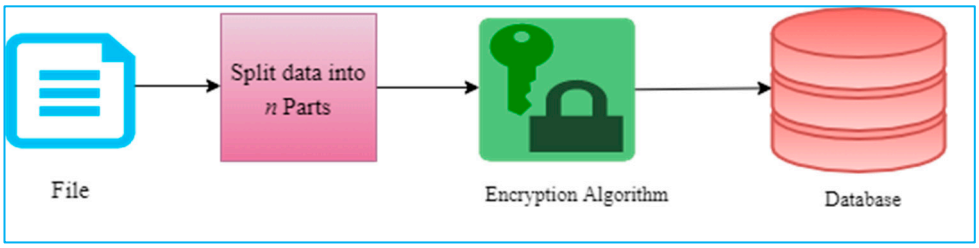

Figure 19. RSA with partitioned file technique [22].

The only problem with this technique is the management of keys of the different blocks.

2.20. Optimization-Based Encryption

Optimization-based encryption [23] uses optimization algorithms to secure the data offloading to the cloud computing system. This works on various smart algorithms, such as Particle Swarm Optimization (PSO), Parallel Particle Swarm Optimization (PPSO), Best-Fit Decrease Algorithm (BFD), Genetic Algorithm (GA), and Ant Colony Optimization (ACO). The mentioned algorithms are used for optimizing results and taking advantage of the efficiency of system resources in the cloud.

2.21. DNA-Based Encryption

The DNA-based encryption technique was proposed in [24], which implies the biological concept of DNA for securing big data in the cloud. As a result of this technique, a 1024-bit secret key based on DNA computing, user attributes, and user's Media Access Control (MAC) address, Supplementary and Decimal Encoding Rule, and the American Standard Code for Information Interchange (ASCII) value is generated. This method allows the system to protect itself against many security attacks.

2.22. Fully Homomorphic Encryption with Advanced Performance

A framework for measuring big data efficiency in terms of precision and performance was introduced in [25]. Fully Homomorphic Encryption (FHE) has been used as a powerful and emerging encryption system capable of performing analytical tasks on encrypted data.

2.23. Advanced Encryption Standard-Based Symmetric Technique

The Cloud Service Providers (CSPs) use AES-based encryption techniques [26] to keep critical customer information, such as credit card numbers and confidential private data. Only authorized customers have the right to access such types of information. Security is about protecting data against unauthorized access to data, while privacy is about protecting the customer's identity.

2.24. Attributes-Based Encryption

Access control mechanisms are linked to the security policies that are provided when customers access cloud services. Attributes-Based Encryption (ABE) controls access to unauthorized cloud data. Typically, a company has security controls that allow employees to access a record rather than giving them full access to the data. Such control is possible through the ABE mechanism that restricts an employee's access to a specific data group. The ABE controls must be configured in cloud projects to prevent unauthorized access.

2.25. Dynamic Security Properties Monitoring Architecture

The dynamic security approach in [27] is based on the concept of virtualization architecture. The primary purpose of this architecture is to identify several threats in the instrumentation of virtualization environments. The main focus in this monitoring approach is how to tackle the device drivers' attacks and supervise applications at run-time.

2.26. Secure Execution Environment for Agents

To provide an auto-configurable environment for mobile agents in ubiquitous computing scenarios, trusted platforms and profile approach is proposed in [28]. This approach is used when an interaction is made with the applications or devices. This security mechanism allows one party to verify the other parties' trustworthiness. The main idea was to create a chain of trust between all elements in the computing environment.

3. Critical Analysis

The discussed encryption techniques for cloud data security are used according to the user requirements; however, there are some common parameters based on which we can compare these techniques to make selection easy for a novice cloud user. All algorithms discussed earlier perform encryption of cloud data in different scenarios. These algorithms are classified broadly into two categories: (1) symmetric and (2) asymmetric encryption techniques. Therefore, ranking any algorithm into low or high classes is unfair. Symmetric encryption techniques use the same keys for encryption and decryption, whereas asymmetric techniques use different encryption and decryption keys. Table 1 compares symmetric and asymmetric techniques based on five commonly used features: keys, speed, text size, key exchange, and many keys.

Table 1. Comparison of symmetric and asymmetric encryption techniques.

Feature	Symmetric Encryption	Asymmetric Encryption
Keys used for encryption/decryption	The same key is used for encryption and decryption	Different keys are used for encryption and decryption
Processing Speed	Fast	Slow
Size of encrypted text	Small or less than original text size	Larger than original text size
Key exchange	Challenging	Not a problem
No keys are required for each participant	Equals to the square of a number of participants	Same as number of participants

Table 2 gives a broad evaluation of the techniques mentioned above based on confidentially, access control, and integrity used for data security in cloud computing research.

Table 2. Comparison of encryption techniques based on security features.

Technique No.	Confidentiality	Access Control	Integrity
2.1	✔	✖	✔
2.2	✔	✔	✖
2.3	✔	✔	✔
2.4	✔	✖	✔
2.5	✔	✔	✔
2.6	✔	✔	✖
2.7	✔	✔	✔
2.8	✔	✖	✔

Table 2. *Cont.*

Technique No.	Confidentiality	Access Control	Integrity
2.9	✔	✔	✖
2.10	✔	✖	✔
2.11	✔	✖	✔
2.12	✔	✖	✖
2.13	✔	✖	✖
2.14	✔	✔	✔
2.15	✔	✖	✖
2.16	✔	✔	✖
2.17	✔	✔	✔
2.18	✔	✖	✔
2.19	✔	✖	✖
2.20	✔	✖	✖
2.21	✔	✔	✖
2.22	✔	✖	✔
2.23	✔	✖	✔
2.24	✔	✔	✔

In Table 3, an overall comparison of the studied techniques is shown based on different features that can help in choosing encryption techniques as per the organization's requirements.

Table 3. Comparison of the encryption techniques based on available features.

Technique No.	Year	Approach Used	Methodology	Encryption End	Encryption Time	Type of Encryption
2.1	2016	Blow Fish	Compress the data file before encryption	Client-side	High	Symmetric
2.2	2014	RSA and AES	RSA is used and then AES	Both on the client-side and server side	Low	Asymmetric
2.3	2013	AES	Data encryption and decryption done by userthemself	Client-side	High	Symmetric
2.4	2014	Fully Homomorphic RSA	Computations can be done on ciphertext without exposing the original data	Server-side	Low	Symmetric
2.5	2014	Private: AES Public: Blowfish Hybrid: Tier1: SAES/IDEA/ Blowfish Tier2: Blowfish then IDEA	Different encryption techniques for public, private, and hybrid sections of the cloud	Server-side	Private: high Public: high Hybrid: high for more secure tier	Asymmetric
2.6	2012	Caesar cipher, RSA and monoalphabetic substitution	Only authorized users can access the data	Server-side	High	Asymmetric

Table 3. *Cont.*

Technique No.	Year	Approach Used	Methodology	Encryption End	Encryption Time	Type of Encryption
2.7	2016	Data with higher privacy value is encrypted by using AES	Only data with A high privacy value is encrypted in certain time constraints	Server-side	High	Symmetric
2.8	2012	RC4, RC6, MARS, AES, DES, 3DES, Blowfish	Best encryption algorithm is chosen according to cloud infrastructure	Server-side	High	Can be according to selected algorithm
2.9	2012	RSA and MD5	Bi-directional security, both at user and cloud admin end	Client-side	Low	Asymmetric
2.10	2010	RSA	RSA generates the digital signature to check message authenticity	Server-side	Low	Asymmetric
2.11	2017	AD2, SED2, EDCon	Split data in two parts and then encrypt both data Split and store in different cloud servers	Server-side	Low	Symmetric
2.12	2019	RSA with CKMS	Encryption is done with RSA and CKMS is used for handling the key	Server-side	Low	Symmetric
2.13	2019	DES, AES, RSA	A combination of three algorithms is used	Server-side	High	Asymmetric
2.14	2013	RSA, DES, AES, Blowfish	Uses different techniques for different aspects	Client-side	High	Asymmetric
2.15	2016	RSA	AWS cloud database is involved	Client-side	High	Symmetric
2.16	2016	RSA, AES	Plain text is encrypted with AES and then RSA is used to encrypt the AES key	Client-side	Low	Both Symmetric and Asymmetric
2.17	2018	AES, Blowfish	Plain text is split into n blocks and then each part is divided into two parts encrypted with a different algorithm using their respective hash key and MD5	Client-side	High	Symmetric
2.18	2018	RSA	Uses RSA with Multi-Party Computation	Client-side	Low	Asymmetric

Table 3. *Cont.*

Technique No.	Year	Approach Used	Methodology	Encryption End	Encryption Time	Type of Encryption
2.19	2018	RSA	Split data into blocks and encrypt each block with their respective key	Server-side	Low	Asymmetric
2.20	2020	PSO, PPSO, BFD, GA, ACO	Optimization procedure	Nil	Nil	Optimization mechanisms
2.21	2020	AES	Media Access Control (MAC) address with key size 1024	Both server and client-side	Low	Symmetric
2.22	2020	Fully Homomorphic Encryption (FHE)	Fully Homomorphic Encryption (FHE) has been used to carry out analytical tasks on encrypted data	Client-side	Low	Symmetric
2.23	2020	AES	CSPs use encryption and other techniques to preserve the privacy of client's critical information	Client-side	High	Symmetric
2.24	2020	Attribute-based encryption	Use of attributes-based encryption (ABE)	Both server and client-side	High	Asymmetric

4. Smart Systems Data Security Challenges in Cloud Computing

Smart systems data security in the cloud computing environment is a challenging problem that leads to the investigation of new mechanisms for achieving a specific level of confidence. The following sub-sections enlist some of the focusing data security challenges.

4.1. Data Integrity and Privacy

Even though cloud computing provides the facility of resource illusion at a lower price, lower resource management, and high availability, it is susceptible to security threats because cloud users are increasing exponentially and the application area hosted in the cloud is exceptionally high. These things cause more significant security threats to cloud purchasers. The associated cloud attack prospering on knowledge entities can result in knowledge breaches and unauthorized access to user information. Due to such integrity violations, cloud knowledge lost its multi-tenant nature; particularly, SaaS suppliers might also have lost their technical knowledge. Virtualizing multiple physical resources among many users results in attacks by malicious insiders of the cloud service provider (CSP) and organization [29]. The shared resources facility enables malicious users to perform attacks on the knowledge of different clients and process their knowledge, hence compromising data privacy and integrity. The other significant risk is outsourcing one's knowledge to third-party storage by the CSP.

The key generation and management for cloud computing cryptography are not standardized and up to the mark. The abnormal and insecure key management prevents quality cryptography algorithms from performing well in generic cloud computing models. Such cryptography might also make sure of the potential risks to cloud computing.

4.2. Private Cloud Security

Security challenges embrace the high value of implementation and management, skills demanded, and vulnerability management. Due to this readying model, the security implementation is typically supported by risk assessment, and thus, the protection cowl is not comprehensive.

4.3. Public Cloud Security

Public cloud security is more challenging than the private cloud because the resources do not seem to be always committed; however, this is leveraged across multiple cloud shoppers. This adds an extra burden of guaranteeing all applications and knowledge accessed on the public cloud and, additionally, must manage the multitude of external influences, such as legislative and knowledge protection.

4.4. Hybrid Cloud Security

In a hybrid cloud, security challenges are comparatively high because the readying model is complicated with a heterogeneous setting, multiple orchestrations, and automation tools. This setting needs extra overhead with any oversight, which leads to necessary risk exposure.

4.5. Security Risks

Organizers in the education sector wish to use cloud services that do not seem to be radically different from those services that are managed inside their centers. The challenges in cloud computing are classified into four main aspects: network, access management, cloud infrastructure, and knowledge security. These are crucial areas that put users' data at risk. To grasp the success in addressing security problems and their challenges in higher academic establishments, one needs to analyze varied aspects of cloud challenges, such as threats, risks, and attack models.

4.6. Network Security

The transmission medium through which the user is hooked up with the cloud infrastructure is vulnerable to security risks. Provisioning of the secure medium prevents the escape of sensitive data throughout the transmission medium. The CSP needs protection to keep knowledge safe from a conventional network-based attack, such as DoS [30], Man-in-the-Middle attack, information processing spoofing, packet sniffing, and port scanning [31].

4.7. Cloud Infrastructure Risks

The most common and essential security challenge cloud systems face may be a lack of VM protection. As a result of multiple VMs set on an equivalent PC, a user cannot place a hardware protection device such as a firewall between them. Another challenge is a dynamic atmosphere where VMs are created, terminated, or affected to a different place mechanically, making it terrible to watch traffic and verify if the attack is accruing.

4.8. Interoperability and Portability

Businesses ought to be able to transfer in and out of the cloud computing paradigms and shift to completely different suppliers. Cloud computing services ought to have the power to operate seamlessly with on-premises IT. According to the IDC survey [30] (taking a sample size 244) conducted in 2008, the seven IT systems and applications that were enjoying cloud facilities were: IT Management Applications (26.2%), Cooperative Applications (25.4%), Personal Applications (25%), Business Applications (23.4%), Applications Development and Preparation (16.8%), Server Capability (15.6%), and Storage Capability (15.5%). This result reveals that organizations still have security/privacy considerations in moving their information onto the cloud. The survey shows that 31.5% of the organization can move their Storage Capability to the cloud in three years. However, this range remains low compared to Cooperative Applications (46.3%) at that point. One of the main reasons for the dearth is the interoperability issue. The portability is also a major concern and bottleneck.

4.9. Performance and Information Measure Cost

Businesses avoid wasting cash on buying hardware; however, they have to pay for information measures. it may be a coffee value for smaller applications but significantly

high for the applications that need a great deal of knowledge. Delivering advanced and exact information over the network needs decent information measures. Due to this, several businesses are still expecting a reduced value before moving to the cloud.

4.10. Possible Solutions

The cloud computing research community focuses on finding possible solutions to tackle the day-to-day new security challenges. In the following sub-sections, we present some initial-level solutions found by the researchers.

4.11. User Authentication

A user on the cloud must be a legitimate user. To verify if there is any inappropriate change in data and information, integrity is the best method [32]. The digital signature approach is used for this type of problem-solving [33]. The approach proposed in [34] is decentralized and robust, in which the cloud serves to identify the ultimate user without knowing their information, which is stored in an encrypted form. The authentic user can only decrypt this information.

4.12. Confidentiality

To prevent the data needed to design a secure data storage system in the cloud, confidentiality can be achieved if the system has deployed a good encryption algorithm and an effective key management system. Attribute-based cryptography [35] is a possible solution with the goal that the client can share information in an adaptable and dynamic way.

4.13. Encryption

The use of an encryption algorithm is the best way to prevent and secure data or information in the cloud computing system. Using a well-designed algorithm provides the best security system. The main issue of using an encryption algorithm increases the computational time. If a double encryption algorithm [36] is used, it increases the computational time exponentially. The other method, called Fully Holomorphic Encryption, can compute results of encrypted information processing rather than raw data, which may build potential information secrecy.

4.14. Data-Centric Approach for Data Loss

If data is not appropriately managed, there may be an issue of data storage and unauthorized access [37]. When data is moved into the cloud, it is justifiable to be worried about its security. Losing information from the cloud, either due to accidental erasure, pernicious altering, or a demonstration of nature cuts down a cloud specialist organization, could be lamentable for an endeavor by a business. Regularly a DDoS assault is just a preoccupation for a more noteworthy danger, for example, an endeavor to take or erase information. For preventing the loss of data, a data-centric approach is devised.

4.15. Key Management

"Poor encryption is bad, but poor key management is worse". To manage the key is the most significant security problem in cloud computing. There is a very complex method to store an encrypted key over the cloud. The solution to this problem is two-level encryption of the used keys [38]. The double encryption technique makes it hard to decrypt the secret information easily.

5. Conclusions and Future Work

Although cloud computing provides benefits, such as resource sharing and on-demand services, provided without spending too much on building the infrastructure and buying resources, many security challenges need to be resolved for offloading smart systems-generated data. Many encryption algorithms and data privacy models have been proposed in the literature to address the arising challenges, but this problem still leaves gaps for

researchers to investigate. While reviewing various encryption algorithms for the data disseminated by various appliances in the smart architecture, it can be concluded that a fine balance between complexity and security needs to be reached when selecting an algorithm dependent on the organization's business needs. No algorithm can be termed the 'best' or 'one-size-fits-all'—each algorithm needs to be studied for its' merits and demerits. A complex algorithm generally takes more time in encryption or decryption, while an algorithm with less complexity may not suit highly confidential data. The Blowfish algorithm stands out amongst symmetric encryption algorithms. When there is a limitation on processing power and time, the AES is the most secure of the symmetric algorithms. The RSA algorithm is an asymmetric algorithm that is suitable where confidential information is to be shared because its public-private key pair security is more ensured. Currently, ongoing research is finding an encryption algorithm that scales well with increasing data generated with high speed by smart systems and is efficient in performance. Data security is the main hindrance due to which smart systems management is hesitant to shift their data to the cloud. Keys used for data encryption and decryption should be more secured so that a third party cannot hack authentication details. By achieving this, we can protect against data tampering. By combing different encryption techniques, data security can be achieved, even though it will increase the encryption and decryption time, and hence, performance is degraded. For maximum throughput, parallel data encryption can be considered in future work.

Author Contributions: Conceptualization, M.B.Q., M.S.Q., S.T., A.A., S.H., M.U. and C.-L.C.; methodology, M.B.Q., M.S.Q., S.T. and A.A.; software, S.H., M.U. and C.-L.C.; validation, M.B.Q., M.S.Q., S.T., A.A., S.H., M.U. and C.-L.C.; formal analysis, M.B.Q., M.S.Q., S.T., A.A., S.H., M.U. and C.-L.C.; investigation, M.B.Q., M.S.Q., S.T., A.A., S.H., M.U. and C.-L.C.; resources, M.B.Q., M.S.Q., S.T., A.A., S.H., M.U. and C.-L.C.; data curation, M.B.Q., M.S.Q., S.T. and A.A.; writing—original draft preparation, M.B.Q., M.S.Q. and S.T.; writing—review and editing, A.A., S.H., M.U. and C.-L.C.; visualization, M.B.Q., M.S.Q., S.T., A.A., S.H., M.U. and C.-L.C.; supervision, M.B.Q., M.S.Q., A.A., S.H., M.U. and C.-L.C.; project administration, M.B.Q., Qureshi M.S and A.A.; funding acquisition, S.H., M.U. and C.-L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Ministry of Science and Technology, Taiwan, under Contract MOST 110-2218-E-305-001–MBK and Contract MOST 110-2410-H-324-004-MY2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the support of Ministry of Science and Technology, Taiwan for paying the Article Processing Charges (APC) of this publication and provision of research facilities.

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

References

1. Mondal, A.; Paul, S.; Goswami, R.T.; Nath, S. Cloud computing security issues & challenges: A review. In Proceedings of the 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 22–24 January 2020; pp. 1–5.
2. Narayan, S.; Gagné, M.; Safavi-Naini, R. Privacy preserving EHR system using attribute-based infrastructure. In Proceedings of the 2010 ACM workshop on Cloud Computing Security Workshop, Chicago, IL, USA, 8 October 2010; pp. 47–52.
3. Grover, A.; Kaur, B. A framework for cloud data security. In Proceedings of the Computing, Communication and Automation (ICCCA), Greater Noida, India, 29–30 April 2016; pp. 1199–1203.
4. Khanezaei, N.; Hanapi, Z.M. A framework based on RSA and AES encryption algorithms for cloud computing services. In Proceedings of the Systems, Process and Control (ICSPC), Kuala Lumpur, Malaysia, 12–14 December 2014; pp. 58–62.
5. Abha, S.; Bhanali, M. Cloud computing security using AES algorithm. *Int. J. Comput. Appl.* **2013**, *67*, 19–23.
6. Zhao, F.; Li, C.; Liu, C.F. A cloud computing security solution based on fully homomorphic encryption. In Proceedings of the 16th International Conference on Advanced Communication Technology (ICACT), Pyeongchang, Korea, 16–19 February 2014; pp. 485–488.

7. Kaur, R.; Singh, R.P. Enhanced cloud computing security and integrity verification via novel encryption techniques. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics, Delhi, India, 24–27 September 2014; pp. 1227–1233.
8. Rani, S.; Gangal, A. Cloud security with encryption using hybrid algorithm and secured endpoints. *Int. J. Comput. Sci. Inf. Technol.* **2012**, *3*, 4302–4304.
9. Gai, K.; Qiu, M.; Zhao, H.; Xiong, J. Privacy-aware adaptive data encryption strategy of big data in cloud computing. In Proceedings of the 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing, Beijing, China, 25–27 June 2016; pp. 273–278.
10. Mohamed, M.E.; Abdelkader, H.S.; El-Etriby, S. Enhanced data security model for cloud computing. In Proceedings of the 8th International Conference on Informatics and Systems (INFOS), Giza, Egypt, 14–16 May 2012; pp. 12–17.
11. Kumar, D.A.; Dubey, A.K.; Namdev, M.; Shrivastava, S.S. Cloud-user security based on RSA and MD5 algorithm for resource attestation and sharing in java environment. In Proceedings of the 2012 CSI Sixth International Conference on Software Engineering (CONSEG), Indore, India, 5–7 September 2012; pp. 1–8.
12. Uma, S.; Lakhani, K.; Mundra, M. Implementing digital signature with RSA encryption algorithm to enhance the data security of cloud in cloud computing. In Proceedings of the 1st International Conference on Parallel Distributed and Grid Computing (PDGC), Solan, India, 28–30 October 2010; pp. 211–216.
13. Li, Y.; Gai, K.; Qiu, L.; Zhao, H. Intelligent cryptography approach for secure distributed big data storage in cloud computing. *Inf. Sci.* **2017**, *387*, 103–115. [CrossRef]
14. Pradeep, K.V.; Vijayakumar, V.; Subramaniyaswamy, V. An efficient framework for sharing a file in a secure manner using asymmetric key distribution management in cloud environment. *J. Comput. Netw. Commun.* **2019**, *2019*, 9852472. [CrossRef]
15. Khan, S.S.; Tuteja, R.R. Data security in cloud computing using cryptographic algorithms. *Int. J. Innov. Res. Comput. Commun. Eng.* **2019**, *7*, 1.
16. Manpreet, K.; Singh, R. Implementing encryption algorithms to enhance data security of cloud in cloud computing. *Int. J. Comput. Appl.* **2013**, *70*, 18.
17. Poteya, M.; Dhoteb, A.; Sharmac, H. Homomorphic encryption for security of cloud data. *Procedia Comput. Sci.* **2016**, *79*, 175–181. [CrossRef]
18. Kartit, Z.; Azougaghe, A.; Kamal Idrissi, H.; Marraki, M.E.; Hedabou, M.; Belkasmi, M.; Kartit, A. Applying encryption algorithm for data security in cloud storage. *Adv. Ubiquitous Netw. Lect. Notes Electr. Eng.* **2015**, *366*, 141–154.
19. Salama, D. Improving the security of cloud computing by building new hybrid cryptography algorithms. *Int. J. Electron. Inf. Eng.* **2018**, *8*, 40–48.
20. Das, D. Secure cloud computing algorithm using homomorphic encryption and multi-party computation. In Proceedings of the International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 10–12 January 2018; pp. 391–396.
21. Hyseni, D.; Luma, A.; Selimi, B.; Cico, B. The proposed model increase security of sensitive data in cloud computing. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 203–210. [CrossRef]
22. Murthy, N.K.; Selvam, R. Security issues and challenges in cloud computing. *Int. Adv. Res. J. Sci. Eng. Technol.* **2015**, *2*, 12.
23. Yu, S.; Wang, C.; Ren, K.; Lou, W. Achieving secure, scalable, and fine-grained data access control in cloud computing. In Proceedings of the IEEE INFOCOM, San Diego, CA, USA, 14–19 March 2010; pp. 1–9.
24. Purushothama, B.; Amberker, B. Efficient query processing on outsourced encrypted data in cloud with privacy preservation. In Proceedings of the International Symposium on Cloud and Services Computing, Mangalore, India, 17–18 December 2012; pp. 88–95.
25. Tebaa, M.; Hajji, S.E.; Ghazi, A.E. Homomorphic encryption method applied to Cloud Computing. In Proceedings of the 2012 National Days of Network Security and Systems, Marrakech, Morocco, 20–21 April 2012; pp. 86–89.
26. Naresh, V.; Thirumala, B. A study on data storage security issues in cloud computing. *Procedia Comput. Sci.* **2016**, *92*, 128–135.
27. Rao, R.; Selvamani, K. Data security challenges and its solutions in cloud computing. *Procedia Comput. Sci.* **2015**, *48*, 204–209. [CrossRef]
28. Wang, G.; Liu, Q.; Wu, J. Achieving fine grained access control for secure data sharing on cloud servers. *Concurr. Comput. Pract. Exp.* **2011**, *23*, 1443–1464. [CrossRef]
29. Gururaj, R.; Mohsin, I.; Farrukh, K. A comprehensive survey on security in cloud computing. *Procedia Comput. Sci.* **2017**, *110*, 465–472.
30. Khalil, A.; Faiz, A.; Mohammad, H.; Hassen, F. Cloud computing security challenges in higher educational institutions—A survey. *Int. J. Comput. Appl.* **2017**, *161*, 22–29.
31. Mahmud, R.; Srirama, S.N.; Ramamohanarao, K.; Buyya, R. Profit-aware application placement for integrated fog–cloud computing environments. *J. Parallel Distrib. Comput.* **2020**, *135*, 177–190. [CrossRef]
32. Hussain, S.; Ullah, S.S.; Uddin, M.; Iqbal, J.; Chen, C.-L. A Comprehensive Survey on Signcryption Security Mechanisms in Wireless Body Area Networks. *Sensors* **2022**, *22*, 1072. [CrossRef]
33. Uddin, M.; Khalique, A.; Jumani, A.K.; Ullah, S.S.; Hussain, S. Next-Generation Blockchain-Enabled Virtualized Cloud Security Solutions: Review and Open Challenges. *Electronics* **2021**, *10*, 2493. [CrossRef]
34. Yanes, A.R.; Martinez, P.; Ahmad, R. Towards automated aquaponics: A review on monitoring, IoT, and smart systems. *J. Clean. Prod.* **2020**, *263*, 121571. [CrossRef]

35. Ma, L.; Wang, X.; Wang, X.; Wang, L.; Shi, Y.; Huang, M. TCDA: Truthful combinatorial double auctions for mobile edge computing in industrial Internet of Things. *IEEE Trans. Mob. Comput.* **2021**. [CrossRef]
36. Munoz, A.; Mana, A.; González, J. Dynamic Security Properties Monitoring Architecture for Cloud Computing. In *Security Engineering for Cloud Computing*; IGI Global: Hershey, PA, USA, 2013; pp. 1–18.
37. Lopez, J.; Mana, A.; Munoz, A. A Secure and Auto-configurable Environment for Mobile Agents in Ubiquitous Computing Scenarios. In Proceedings of the Third International Conference on Ubiquitous Intelligence and Computing, Wuhan, China, 3–6 September 2006; Volume 4159, pp. 977–987.
38. Hussain, S.; Ullah, I.; Khattak, H.; Adnan, M.; Kumari, S.; Ullah, S.S.; Khan, M.A.; Khattak, S.J. A lightweight and formally secure certificate based signcryption with proxy re-encryption (CBSRE) for Internet of Things enabled smart grid. *IEEE Access* **2020**, *8*, 93230–93248. [CrossRef]