Sergey Y. Yurish
Editor

Advances in Networks, Security and Communications

Volume 3



Sergey Y. Yurish
Editor

Advances in Networks, Security and Communications

Volume 3

International Frequency Sensor Association Publishing

Sergey Y. Yurish
Editor

Advances in Networks, Security and Communications
Volume 3

Copyright © 2022 by International Frequency Sensor Association Publishing, S. L.

E-mail (for orders and customer service enquires): ifsa.books@sensorsportal.com

Visit our Home Page on http://www.sensorsportal.com

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (IFSA Publishing, S. L., Barcelona, Spain).

Neither the authors nor International Frequency Sensor Association Publishing accept any responsibility or liability for loss or damage occasioned to any person or property through using the material, instructions, methods or ideas contained herein, or acting or refraining from acting as a result of such use.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identifies as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

e-ISBN: 978-84-09-47239-0

BN-20221221-XX

BIC: TJK

Contents

Contributors **11**

Preface **13**

1. Novel Results on Blind Symbol Rate Estimation in Digital Satellite Receivers **15**
1.1. Introduction 15
1.2. Signal Model 16
1.3. Cramer-Rao Lower Bound 17
1.3.1. Computation of the Modified CRLB 17
1.3.2. Computation of the Asymptotic CRLB 17
1.4. Estimator Algorithms 20
1.4.1. CCB and ML Approach 20
1.4.2. DF Concept 21
1.5. Numerical Results 23
1.6. Conclusions 26
Acknowledgements 26
References 26
Appendix 28

2. Confidentiality Concerns in Cloud Environments **29**
2.1. Introduction 29
2.2. Overview of Existing Attacks in Cloud Computing 30
2.2.1. Authentication Attacks 30
2.2.2. Man-in-the-middle Attacks 31
2.2.3. Denial of Service Attacks 32
2.2.4. Cloud Malware Injection Attacks 33
2.2.5. Side Channel Attacks 33
2.2.6. Collusion Attacks 33
2.3. Data Confidentiality in Cloud Computing 34
2.3.1. Classification of Solutions Ensuring Data Confidentiality in Cloud Computing 34
2.3.2. Confidentiality Through Biometric Encryption 36
2.3.2.1. Biometric Encryption in Mobile Cloud Computing *36*
2.3.2.2. Privacy-preserving Biometric Identification in Cloud Computing *36*
2.3.2.3. CloudBI *37*
2.3.2.4. BE to Enhance Confidentiality in Cloud Computing *38*
2.3.2.5. Privacy Preserving Security Using Biometrics *38*
2.3.3. Confidentiality Through Homomorphic Encryption 39
2.3.3.1. Cloud El Gamal and Fast Cloud RSA Schemes *39*
2.3.3.2. Fast-variant Cloud Paillier Scheme *40*
2.3.3.3. A Solution Based on Fully Homomorphic Encryption *40*
2.3.3.4. Speedy Cloud-RSA *41*
2.3.3.5. Solution Based on Intel SGX *41*
2.3.4. Confidentiality not by Encryption only 42
2.3.4.1. HPI-Secure *42*
2.3.4.2. K-NN Classifier for Data Confidentiality *43*
2.3.4.3. Steganography-based Solution *44*
2.3.4.4. Confidentiality Preserving Through Encryption and Obfuscation *44*

2.3.4.5. Attribute-based Proxy Re-encryption ... *45*
2.3.4.6. TR-MABE ... *46*
2.3.4.7. Privacy-manager ... *47*
2.3.4.8. Hierarchical Classes of Computing for Privacy Preserving ... *48*
2.3.4.9. Encryption and Trust-based Solution ... *49*
2.3.4.10. Blockchain for Cloud-based Data Privacy ... *50*
2.3.4.11. L-EncDB ... *52*
2.3.4.12. CS-IBE ... *53*
2.3.4.13. GRBAC ... *53*
2.3.4.14. ISRSAC ... *54*
2.3.4.15. Solution Using Modular Encryption Standard ... *55*
2.3.4.16. CPDA ... *56*
2.3.5. Confidentiality According to the Type of Storage ... 57
2.3.5.1. Secret Sharing Scheme ... *57*
2.3.5.2. Scheme for Image Data Privacy ... *57*
2.3.5.3. DCSSP ... *58*
2.3.5.4. CAFE ... *59*
2.3.5.5. ARIANNA ... *60*
2.3.6. Confidentiality Without Encryption ... 60
2.3.6.1. NetDB2-MS ... *60*
2.3.6.2. Multiple Storage Solution ... *61*
2.3.6.3. MONcrypt ... *62*
2.3.6.4. Matrix-based Solution ... *63*
2.4. General Assessment of the Studied Solutions ... 63
2.5. Conclusion ... 67
References ... 68

3. BAC-DC: Blockchain-based Access Control for Decentralized Cloud Data ... 71
3.1. Introduction ... 71
3.2. Background and Related Works ... 72
3.2.1. Meta-key ... 72
3.2.2. Storj ... 73
3.2.3. Blockchain technology for Data Privacy in cloud computing ... 75
3.2.4. BACC (Blockchain-based Access Control for Cloud Data) ... 75
3.2.5. FileCoin ... 76
3.2.6. SIA 77
3.2.7. Discussion ... 77
3.3. Overview of BACC (Blockchain-based Access Control for Cloud Data) ... 79
3.3.1. The Main Components of the System ... 79
3.3.2. System Configuration ... 80
3.3.2.1. Data Upload ... *81*
3.3.2.2. Data Download ... *81*
3.3.3. The Limits of the BACC Scheme ... 81
3.4. Transition from Centralized to Decentralized Storage ... 83
3.4.1. Basic Cloud Computing ... 83
3.4.2. Why Decentralizing the Cloud? ... 83
3.4.3. Decentralized Cloud ... 84
3.5. Proposed Model BAC-DC: Blockchain-based Access Control for Decentralized Cloud Data ... 85
3.5.1. The Main Components ... 85
3.5.1.1. The Nodes ... *85*
3.5.1.2. The Peer-to-peer Communication and Discovery System ... *85*
3.5.1.3. The Reference Table or Metadata ... *86*

3.5.2. BAC-DC: Operating Mode....86
3.5.2.1. Data Upload....86
3.5.2.2. Data Download....87
3.5.3. Model Architecture....88
3.6. Link between Our BAC-DC Scheme and the One Studied BACC....89
3.7. Security Analysis....90
3.7.1. File Partitioning Before Storage....90
3.7.2. End-to-end Encryption....91
3.7.3. Parity Fragments for Redundancy....92
3.7.4. File Verification....92
3.7.5. Data Loss....92
3.7.6. Access Control....92
3.8. Security against Attacks....94
3.8.1. The Google Attack....94
3.8.2. The Sybil Redundancy Attack....95
3.8.3. The Incorrect Distribution....95
3.8.4. Cheating Client Node....95
3.8.5. The Hostage Byte Attack....95
3.8.6. Geppetto's Honest Attack....96
3.9. Advantages and Disadvantages of the BAC-DC Model....96
3.9.1. Advantages....96
3.9.2. Disadvantages....97
3.10. Conclusion....98
References....98

4. Attack Detection in IoT Networks Using Intelligent Classification Mechanisms....101
4.1. Introduction....101
4.2. Security Issues in IoT Networks....102
4.2.1. Physical Threats....102
4.2.2. Cyber Threats....103
4.2.2.1. Denial of Service: DoS / DDoS....103
4.2.2.2. Mirai....103
4.2.2.3. R2L (Remote to Local)....103
4.2.2.4. U2R (User to Root)....104
4.2.2.5. Man-in-the-middle....104
4.2.2.6. Injection Attack....104
4.2.2.7. Password Attack....104
4.3. Defense Solutions Associated with an IoT System....105
4.3.1. Network Based Intrusion Detection System (NIDS)....105
4.3.2. Host Based Intrusion Detection System (HIDS)....106
4.4. Anomaly Detection by Classification....106
4.4.1. Data Classification in AI....107
4.4.2. Classification Types....107
4.4.3. Classification Algorithms....108
4.4.3.1. Linear Models....108
4.4.3.2. Nonlinear Models....109
4.4.4. Anomaly Detection by Binary Classification....109
4.4.5. Detection and Identification of Anomalies by Multiclass Classification....110
4.5. Presentation of Used Tools, Datasets and Methods....111
4.5.1. Tools and Libraries....111
4.5.2. Dataset 1: UNSW-NB15....113
4.5.3. Dataset 2: IoT Network Intrusion Dataset....116
4.5.4. Method....117

4.5.5. Data Pre-processing 118
4.5.5.1. Shape Analysis 118
4.5.5.2. Background Analysis 121
4.5.5.3. Data Preparation 124
4.6. Results and Discussion 125
4.6.1. Evaluation Metrics 125
4.6.1.1. Confusion Matrix 125
4.6.1.2. Metrics 126
4.6.2. Analysis and Interpretation of Results 126
4.6.2.1. Anomaly Detection 126
4.6.2.2. Identification of Attacks 132
4.7. Conclusion 137
References 139

5. Emerging Trends Using IoMT for Healthcare System 141
5.1. Introduction 141
5.2. Literature Review 142
5.3. General Framework of IOMT 142
5.4. IoMT Conceptual Frameworks and Architecture 143
5.5. Technologies & Platforms behind IoMT 144
5.5.1. Device Integration Layer 146
5.5.2. Information Integration Layer 146
5.5.3. Information Extraction or Information Representation Layer 146
5.6. IoMT Communication Technologies 146
5.7. Typical IoMT-based E-health System 147
5.8. Real Time Health Monitoring Systems 149
5.9. Cloud, Fog & Edge Computing 150
5.10. Security Aspects in IoMT Communications 151
References 153

Index 157

Contributors

Cherif Diallo
Laboratoire d'Algèbre, Cryptographie, Codes et Applications (LACCA),
Dept. Informatique, UFR SAT, Université Gaston Berger, BP 234, Saint-Louis, Sénégal
E-mail: cherif.diallo@ugb.edu.sn

Wilfried Gappmair
Institute of Communication Networks and Satellite Communications,
Graz University of Technology, Austria

P. Madhavasarma
SASTRA Deemed University, Thanjavur 613401, Tamilnadu, India
E-mail: spms1975@yahoo.co.in

T. N. Prabakar
Sri Sairam College of Engineering, Bangalore, India

Barbara Suesser-Rechberger
Institute of Communication Networks and Satellite Communications,
Graz University of Technology, Austria

P. Veeraragavan
University College of Engineering, Tindivanam 604001, Tamilnadu, India

Sarah Wayzani
Laboratoire d'Algèbre, Cryptographie, Codes et Applications (LACCA),
Dept. Informatique, UFR SAT, Université Gaston Berger, BP 234, Saint-Louis, Sénégal

Preface

After successful publication of the first two volumes of '*Advances in Networks, Security and Communications'* open access Book Series in 2017 (Volume 1) and in 2019 (Volume 2), and positive feedback from our authors and readers, we have decided to publish the 3rd volume of this Book Series in 2022.

This volume contains 5 chapters written by 7 authors from 3 countries: Austria, India and Senegal. But it is not a simple set of reviews. Each of the chapters contains the extended state-of-the art followed by new, obtained by the authors results, unpublished before.

Chapter 1 presents in details the asymptotic Cramer-Rao lower bound (ACRLB) for carrier-blind non-data-aided (NDA) estimation of the symbol rate. This figure of merit represents the theoretical limit of the jitter variance produced by any estimator in the lower signal-to-noise ratio (SNR) range. It could be observed that the SNR value characterizing the intersection of ACRLB and modified Cramer-Rao lower bound (MCRLB) might be helpful in practice. Furthermore, a novel estimator algorithm has been discussed, which suffers only marginally from error floor effects, although this solution requires a second filter operated in parallel to the receiver matched filter.

The confidentiality concerns in cloud environments are considered in the Chapter 2. The recent solutions concerning this subject have been described by the authors. This will allow to identify a better perspective that will help to develop in future an effective cloud strategy based on existing solutions in order to counter the security challenges regarding data confidentiality.

The blockchain-based access control for decentralized cloud data proposed in Chapter 3 brings many advantages in terms of security. Authors have integrated the blockchain technology into the cloud while emphasizing access control managed by specific nodes called master nodes and not by a central entity. Indeed, it responds to security vulnerabilities identified in existing decentralized cloud computing solutions. Among the many advantages, it is very profitable, emphasizes data confidentiality, offers good redundancy and unlimited bandwidth.

Chapter 4 describes the models and techniques for intelligent classification of network traffic in order to increase accuracy in the anomaly detection and attack identification in IoT environments. The implementation of several artificial intelligence classification models namely a decision tree, random forest, AdaBoost and an artificial neural network for anomaly detection and attack identification in IoT networks have been analyzed and discussed.

Chapter 5 is focused on the emerging trends using Internet of Medical Things (IoMT) for healthcare system. The frameworks, architectures, technologies and platforms of IoMT on the different integration levels are described in the Chapter.

Every chapter in '*Advances in Networks, Security and Communications'* Vol. 3 is independent and self-contained. The book can be useful for post-graduate students, researchers, engineers and scientist working in the appropriate areas.

I shall gratefully receive any advices, comments, suggestions and notes from readers to make the next volumes of '*Advances in Networks, Security and Communications*'open access book Series very interesting and useful.

Dr. Sergey Y. Yurish

Editor
IFSA Publishing *Barcelona, Spain*

Chapter 2
Confidentiality Concerns in Cloud Environments

Sarah Wayzani and Chérif Diallo

2.1. Introduction

Cloud computing as shown in Fig. 2.1, is an "on-demand" availability of a computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user. The cloud is composed of five essential characteristics, three service models and four deployment models [1].

In other words, the cloud computing is the ability to access your work environment, data, software and business applications from any internet connection on any connected device. It is the promise of resource sharing in a pure plug and play model that dramatically simplifies infrastructure planning. The two key advantages (among many others) of this model are ease of-use and cost-effectiveness.

In collaboration with technology consulting firm CITO Research, cloud management consulting firm Commvault surveyed 100 IT professionals and the resulting report states that 93 of them use the cloud in their businesses and have even increased the budget allocated to this technology [2].

However, despite the significant advances that have taken place in recent years and its growing influence, concerns regarding cloud computing still remain. Indeed, companies still have suspicion about the risk of moving from traditional IT to cloud computing environments. Especially with the latest research from ISC2 showing that 93 % of organizations are moderately or extremely concerned about cloud security, and one in four organizations has confirmed a cloud security incident in the past 12 months. [3].

Sarah Wayzani
Laboratoire d'Algèbre, Cryptographie, Codes et Applications (LACCA), Dept. Informatique, UFR SAT, Université Gaston Berger, Saint-Louis, Sénégal

Thus, in the cloud sector, security in general and the confidentiality of personal data in particular are still among the obstacles to adoption. Even if most companies use Cloud technologies and consumers are increasingly adopting dematerialized products and services, it is still very difficult to know exactly if the data stored online is secure. However, the benefits outweigh somehow the drawbacks but the model is worth exploring to understand better his uses.

In order to identify the main privacy issues in cloud environments, we have collected some recent solutions concerning this subject and cited in reference, in order to study their strengths and weaknesses. This will allow us to identify a better perspective that will help us achieve our main objective, which is to develop in future works an effective cloud strategy based on existing solutions in order to counter the security challenges regarding data confidentiality.

Thus, to carry out our work, our mission is first to submit a classification of the solutions collected, then to identify comparison criteria that can help us identify the advantages and disadvantages of each solution category and finally propose tables comparative to support the image of a possible released perspective.

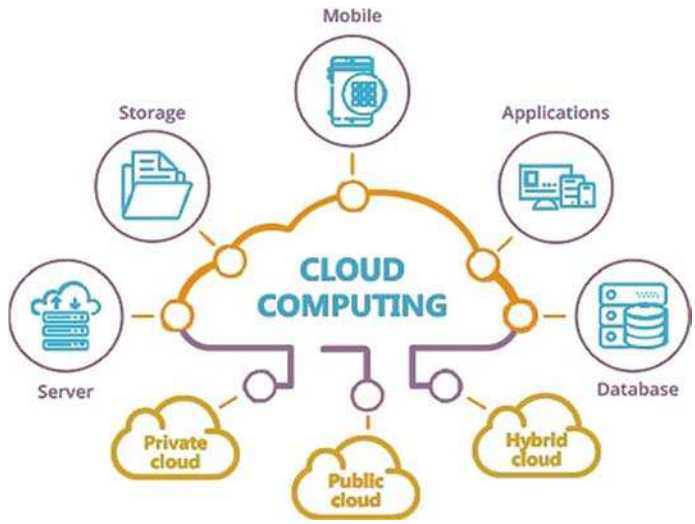


Fig. 2.1. Conceptual view of cloud computing.

2.2. Overview of Existing Attacks in Cloud Computing

2.2.1. Authentication Attacks

Research studies reveal that any authentication mechanism related to web applications and cloud should provide high security, easy to use interface and support user mobility. The customers prefer to access their applications from different locations and different devices such as desktop, laptop, PDA, smart phones, cell phones etc. Those needs pose

significant requirements to the security of applications. The broad range of user requirements introduces wide range of attack vectors in the cloud that makes the security of cloud applications a thought-provoking matter. Cloud service providers need to ensure that only legitimate user is accessing their services and these points out to the requirement of a strong user authentication mechanism. But there exist numerous attacks that can create loop holes in the authentication mechanism and hence identifying the most secure authentication mechanism with high user acceptability is a big challenge in the cloud environment [4]. There are several authentication attacks, among which we can cite:

- Cipher text attack: the attacker has access only to a number of encrypted messages. He has no idea what the plaintext data or the secret key may be. The goal is to recover as much plaintext messages as possible or (preferably) to guess the secret key;
- Eavesdropping attack: this attack takes advantage of unsecured network communications to access the data sent and received;
- Brute force attack: In this type of attack, all possible combinations of password apply to break the password. The brute force attack is generally applied to crack the encrypted passwords where the passwords are saved in the form of encrypted text;
- Dictionary attack: is a method of breaking into a password-protected computer or server by systematically entering every word in a dictionary as a password;
- Known plaintext attack: the attacker has access to both the plaintext (called a crib), and its encrypted version (cipher text);
- Replay attack: is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed;
- Key loggers: The key loggers are the software programs which monitors the user activities by recording each and every key pressed by the user.

2.2.2. Man-in-the-middle Attacks

This type of attacks happens, when a communication established in two-node or computer system. Usually, the attackers modify the message content and message sequences. A man-in-the-middle attack allows a malicious actor to disrupt, send and receive data between the two users. Some types of Man-in-the-middle attacks are:

- Address Resolution Protocol Communication (ARP): In the normal ARP communication, the host PC will send a packet, which has the source and destination IP address inside the packet and will broadcast it to all the devices connected to the network. The device that has the target IP address will only send the ARP reply with its MAC address in it and then communication takes place. The ARP protocol is not a secured protocol and the ARP cache does not have a foolproof mechanism, which causes a major problem.
- ARP Cache Poisoning: In ARP cache poisoning, the attacker would be sniffing onto the network by controlling the network switch to monitor the network traffic, spoof the ARP packets between the host and the destination PC, and perform the MITM attack.

- DNS Spoofing: DNS spoofing, also referred to as DNS cache poisoning, is a form of computer security hacking in which corrupt Domain Name System data is introduced into the DNS resolver's cache, causing the name server to return an incorrect result record, e.g., an IP address. This results in traffic being diverted to the attacker's computer (or any other computer). This is a kind of online MITM attack where the attacker has created a fake website of your bank, so when you visit your bank website you will be redirected to the website created by the attacker and then the attacker will gain all your credentials.
- Session Hijacking: In this, once the session is established between the host PC and the web server the attacker can obtain certain parts of the session establishment that is done by capturing the cookies that were used for the session establishment.

2.2.3. Denial of Service Attacks

In this type of attacks, the attackers send enormous number of requests to the target servers to attack it. Hence, it cannot process the additional requests. DoS attacks are of many types:

- An attacker can overload the target with large amount of junk data that consume the network bandwidth and resources, for example UDP floods, ICMP floods etc.;
- An attacker can make use of blank space (lacuna) that associated with various networking protocol to overload target resource, for example SYN floods, fragment packet attack, ping of death etc.;
- An attacker can make HTTP request in large amount so that it cannot be handle by the server for example HTTP-DDOS attack, XML-DDOS attack, etc.

Denial-of-service (DoS) attacks are one of the major security challenges in the emerging cloud computing models. Currently, numerous types of DoS attacks are conducted against the various cloud services and resources, which target their availability, service level agreements, and performance.

In cloud computing, a DoS attack can be described as an attack designed to prevent some cloud computing service or resource from providing its normal services for a period. DoS attacks compromise the availability of the cloud resources and services and often target the computer networks' bandwidth or connectivity. Generally, DoS attacks come in the following categories:

- Bandwidth attacks;
- Connectivity attacks;
- Resource exhaustion;
- Limitation exploitation;
- Process disruption;

– Data corruption;
– Physical disruption.

2.2.4. Cloud Malware Injection Attacks

In Cloud Malware Injection Attack, an attacker tries to inject malicious service or virtual machine into the cloud. In this type of attack attacker creates its own malicious service implementation module (SaaS or PaaS) or virtual machine instance (IaaS), and try to add it to the Cloud system. Then, the attacker has to behave to make it a valid service to the Cloud system that it is some new service implementation instance among the valid instances. If the attacker succeeds in this, the Cloud automatically redirects the requests of valid user to the malicious service implementation, and the attacker code starts to execute. The main scenario behind the Cloud Malware Injection attack is that an attacker transfers a malicious service instance into cloud so that it can achieve access to the service requests of the victim's service. To achieve this, the attacker has to derive control over the victim's data in the cloud. According to classification, this attack is the major representative of exploiting the service-to-cloud attack surface. The purpose of "cloud malware injection attack" can be anything in which an attacker is interested; it may include data modifications, full functionality changes/reverse or blockings.

2.2.5. Side Channel Attacks

An attacker attempts to compromise the cloud system by placing a malicious virtual machine in close proximity to a target cloud server system and then launching a side channel attack. Side-channel attacks have emerged as a kind of effective security threat targeting system implementation of cryptographic algorithms. Evaluating cryptographic systems resilience to side-channel attacks is therefore important for secure system design. Side channel attacks use two steps to attack:

– VM CO-Residence and Placement i.e., an attacker can often place his or her instance on the same physical machine as a target instance;
– VM Extraction i.e., the ability of a malicious instance to utilize side channels to learn information about co-resident instances.

It can be very easy to gain secret information from a device so security against side channel attack in cloud computing should be provided.

2.2.6. Collusion Attacks

Collusion attack can be defined as the execution of operations that have the ability to combine multiple copies of the media or other files together to produce a new copy. The operations which include but are not limited to are averaging of the data, replacing, linear combination of data, etc. They are most of the time used for breaking into video fingerprinting technologies and cracking the passwords/keys of the required system [5].

In the cloud storage situation, any revoked user can use such operations and can guess the private key easily to break into the system and access the files. Therefore, this problem has to be solved to increase the security of the cloud storage system, maintaining the Integrity of the Specifications.

2.3. Data Confidentiality in Cloud Computing

From initial concept building to current deployment, cloud computing is growing more and more mature. The adoption of cloud computing may lead to several privileges provided by cloud providers including reducing costs, powerful computations, high services elasticity and scalability.

In current cloud computing systems, because users' data are stored and processed by computing systems managed and operated by various service providers, users are concerned with the risks of unauthorized usage of their sensitive data by various entities, including service providers. Concerns over the confidentiality of sensitive data are still then the main obstacles limiting the widespread adoption of cloud computing services. Confidentiality refers to protecting information from being accessed by unauthorized parties. In other words, only the people who are authorized to do so can gain access to sensitive data.

Since users' sensitive data are mostly presented in unencrypted forms to remote machines owned and operated by third party service providers, the risks of unauthorized disclosure of the users' sensitive data by service providers may be quite high. There are many techniques for protecting users' data from outside attackers, but currently no effective way is available for protecting users' sensitive data in cloud computing [9]. From the consumers' perspective, cloud computing security concerns, especially data security and privacy protection issues, remain the primary inhibitor for adoption of cloud computing services [5].

2.3.1. Classification of Solutions Ensuring Data Confidentiality in Cloud Computing

Classification is the process that can allow us to categorize all collected solutions concerning data confidentiality issues in cloud computing environment. Our choice has been to categorize them according to their way to ensure confidentiality in such environment. We ended up choosing five different categories, which are:

- Confidentiality through biometric encryption;
- Confidentiality through homomorphic encryption;
- Confidentiality not by encryption only;
- Confidentiality according to the type of storage;
- Confidentiality without encryption.

Table 2.1 classifies some solutions ensuring data confidentiality in cloud computing according to these five different categories cited above

Table 2.1. Classification of solutions ensuring data confidentiality in cloud computing.

Classification	Solutions
Confidentiality through biometric encryption	Biometric Encryption in mobile cloud computing [6], Privacy preserving biometric Identification in cloud computing [7], CloudBI [8], BE to enhance confidentiality in cloud computing [9], Privacy preserving security using biometrics [10]
Confidentiality through homomorphic encryption	Cloud El Gamal and Fast cloud RSA schemes [11], Fast variant cloud-Paillier scheme [12], A solution based on fully homomorphic encryption [13], Speedy Cloud-RSA [14], Solution based on Intel SGX [15]
Confidentiality not by encryption only	HPI-Secure [16], K-NN classifier for data confidentiality [17], Steganography based solution [18], Confidentiality preserving through encryption and obfuscation [19], Attribute Based Proxy Re-encryption [20], TR-MABE [21], Privacy-Manager [24], Hierarchical classes of computing for privacy-preserving in cloud computing [25], Encryption and trust-based solution [26], Blockchain for cloud-based data privacy [27], L-EncDB [28], CS-IBE [29], GRBAC [30], ISRSAC [31], Solution using Modular Encryption Standard [32], CPDA [33]
Confidentiality according to the type of storage	Secret sharing scheme [36], Scheme for image data privacy [37], DCSSP [38], CAFE [39], ARIANNA [40]
Confidentiality without encryption	NetDB2-MS [42], Multiple storage solution [43], MONcrypt [44], Matrix-based solution [45]

2.3.2. Confidentiality Through Biometric Encryption

2.3.2.1. Biometric Encryption in Mobile Cloud Computing

The authors in [6] try to integrate BE technology in their MCC platform. The steps towards using BE MCC platform are as follows: BE application developers should register in the platform when they release their products. Users download registered applications to mobile devices from app market in the platform, and the biometric data will be transferred to the platform through specific interface. Before a user begins to use the applications, he should create a record containing his biometric features on the platform. In the platform, thousands of requests are scheduled and analyzed, and the original templates of users are stored in database encrypted. After the requests are disposed by CAC (Cloud Authentication Center), the results will be sent back to the applications. At design time, the developer has to register the application with BE MCC platform. Application registration informs request parameters, including the category of the application, biometric features requested, security requirement level etc. The applications will be examined strictly in case of malwares disguising legal software, then the application will have a record data in the platform containing its information.

Discussion: Statistics show that 10 % of human's fingers cannot be identified by biometric feature system due to the quality of fingerprint, and other fingers may become dry, wet or distorted and fail to be collected. Moreover, fingerprint image changes with many environmental or biological factors. Therefore, improving the robustness of fingerprint identification algorithm to deal with fingerprints of low quality is a major problem in this area. In addition, aging, illness, etc., have a bearing on the quality and variability of captured biometric data. Therefore, a comprehensive mechanism is required to help those people keep their biometric features available by some methods, such as multiple fingers, recording or updating the biometric data combining user's other authenticated information. The authors point out several obstacles to overcome in implementing BE in MCC, though some of them are usual problems inherited from BE or MCC, they still need to be solved:

- Extending battery lifetime;
- Improving data storage capacity and processing power;
- Improving reliability.

2.3.2.2. Privacy-preserving Biometric Identification in Cloud Computing

In [7], the authors propose a novel privacy-preserving biometric identification scheme which achieves efficiency by exploiting the power of cloud computing. In their proposed scheme, the biometric database is encrypted and outsourced to the cloud servers. To perform a biometric identification, the database owner generates a credential for the candidate biometric trait and submits it to the cloud. The cloud servers perform identification over the encrypted database using the credential and return the result to the owner. During the identification, cloud learns nothing about the original private biometric

data. Because the identification operations are securely outsourced to the cloud, the real time computational/ communication costs at the owner side are minimal.

Discussion: The proposed scheme introduces negligible computation task to the client side, which is extremely efficient for most mobile devices to implement. From the experimental results, they can clearly see that:

- The scheme can efficiently deal with large size databases in biometric data identification thanks to the high scalability of cloud;
- It can also efficiently handle a relatively large number of identification requests even for large database cases; however, such a setting will cause high delays in previous works;
- Compared with previous works, the scheme just requires lightweight hardware for both database owner side and the client side, which makes our scheme to be easily implemented with a relatively low cost, which further makes it suitable for a wide range of applications.

2.3.2.3. CloudBI

The authors of [8] investigate the privacy-preserving biometric identification outsourcing problem, where the database owner outsources both the large-scale encrypted database and the computationally intensive identification job to the semi-honest cloud, relieving itself from data storage and computation burden. They present new privacy preserving biometric identification protocols, which substantially reduce the computation burden on the database owner. Privacy-preserving biometric identification has been extensively investigated in the secure two-party computation model, where the database owner and the user interactively execute the identification protocol without revealing the self-biometric data information to each other. These works, however, either have efficiency issues (heavily rely on homomorphic encryption) or fail to support the computation of a global minimum, which limits their applications. In this work, they, for the first time, identify the deficiencies and security weaknesses of previous privacy-preserving biometric identification protocols in the computation outsourcing model. Their design is carefully tuned to meet the security and efficiency requirements under a three-party outsourcing computation model, where one or two parties may be semi-honest.

In this paper, the authors consider a cloud-based biometric-matching system involving three parties:

- The database owner;
- The user;
- The cloud server.

Discussion: The results in [8] clearly validate the scalability of the cloud-based privacy-preserving biometric identification system. To demonstrate the computation

savings on the database owner, the authors show the comparison of time costs of biometric identification on the database owner with and without identification outsourcing. As can be seen, the scheme achieves constant time cost on the database owner, while the time cost of performing identification locally (over plaintext biometric database without outsourcing) increases linearly with the database size. The larger the database size (e.g., with 10 million Finger Codes) is, the higher efficiency gain can be achieved.

2.3.2.4. BE to Enhance Confidentiality in Cloud Computing

In [9] the authors proposed biometric encryption to improve the confidentiality in Cloud computing for biometric data. In addition, they discussed virtualization for Cloud computing, as well as biometrics encryption. Indeed, they overviewed the security weaknesses of Cloud computing and how biometric encryption can improve the confidentiality in Cloud computing environment.

Biometric identification is a more flexible authentication method than the secret key, because it is extremely difficult to be forgotten or even lost. It is a solution that has been used to protect biometric identification. Indeed, cryptography was the solution to overcome the threats on biometric identification.

Discussion: Despite all those charming features and strong support that virtualization technology provides to the computing environment that utilizes it, it has a huge impact on the development of these relying environments. Although virtualization is not new technology as Cloud, it has several security issues. These security issues have migrated into Cloud computing environment. Most of these security issues are based on the virtualization level. However, Biometric encryption provides extra level security for privacy against these security issues. It is important to mention that BE does not solve the security issues in Cloud computing. But it secures Biometric data against most of these security issues. BE provides extra level of security for the attacker in the Cloud which is the encryption mechanism.

2.3.2.5. Privacy Preserving Security Using Biometrics

The paper [10] emphasizes on cloud security and privacy issues and provides the solution using biometric face recognition. The authors propose a biometric face recognition approach for security and privacy preservation of cloud users during their access to cloud resources.

In [10], biometric facial feature of individuals is used as key attributes to access the resources of cloud by matching and recognizing the users in the cloud database. The identification is performed on the encrypted data by the cloud servers and returns the matching scores as a result for authentication of owners. The stored data consists of individual records. The storage of personal record as database provides the adequate foundation to build the encrypted biometric database.

Discussion: The major advantages of biometrics over traditional authentication systems (e.g., passwords, and token-based systems) are:

- Higher level of security of stored data;
- Privacy preservation of users;
- Diminutive chances of forgery;
- Cost-effective based solutions;
- User friendliness.

In the proposed recognition system, privacy preservation of users and security of cloud data are preserved based on biometric features of different face images. The features of face images are extracted by applying the Eigen faces recognition algorithm. The proposed biometrics-based recognition system maintained the preservation of user's privacy by performing the identification of individuals based on encrypted biometric templates of face images.

2.3.3. Confidentiality Through Homomorphic Encryption

2.3.3.1. Cloud El Gamal and Fast Cloud RSA Schemes

In [11] all the proposed schemes support multiplicative homomorphism (MH) on integers. The purpose of this article is to compare the Cloud-ElGamal scheme with the CloudRSA schemes. The concern to see sensitive data processed clearly in the cloud computing environment remains the main obstacle to the widespread adoption of cloud services. Using certain techniques to perform calculations on encrypted data seems to be an effective way to overcome this obstacle and build confidence in cloud solutions. In fact, scientists have highlighted a promising technique, Homomorphic Encryption (HE), which offers a third party the ability to perform operations on encrypted data (i.e., capable of ensuring the confidentiality of storage and the processing of data outsourced in the Cloud Computing (CC) environment). Homomorphic property makes homomorphic encryption schemes useful in a number of privacy-preserving applications, such as electronic voting and digital healthcare.

Discussion: Four encryption schemes have been studied in this article supporting homomorphic encryption on integers, namely CloudRSA, MultiPrime Cloud-RSA, MultiPower Cloud-RSA and Cloud-ElGamal. These schemas are very resistant to privacy attacks. The Cloud-RSA scheme and its variants, MultiPrime Cloud-RSA and MultiPower Cloud-RSA, are secure to take into account large composite integers and to discover the decryption exhibitors when the encryption exhibitors are private; while the Cloud-ElGamal scheme is secured by the difficulty of suppressing discrete logarithms on finite groups when the generator selected from the group is private. The Cloud-RSA variants are designed to achieve faster decryption. The simulation results in the article show that the MultiPower Cloud-RSA system offers high decryption speed compared to the three HE systems offered while providing a recommended level of security.

2.3.3.2. Fast-variant Cloud Paillier Scheme

In [12], the authors boosted a Paillier's scheme at security level in order to work efficiently in Cloud computing environments. This proposed scheme is called a fast variant of the Cloud-Paillier encryption scheme. Thus, it addresses an exception of Paillier's scheme, because it supports the additive homomorphism (AH) over the integers and resists more to confidentiality attacks. The scheme is secure under double trapdoor assumptions: the hardness of factoring large composite integers and the hard decisional composite residuosity assumption (HDCRA).

Discussion: The fast Cloud-Paillier's variant keeps the same security level of the Cloud-Paillier scheme. The security of the variant depends on double hard problems:

- The hardness of factoring a composite integer

$$n = pq, \tag{2.1}$$

- The Hard Decisional Composite Residuosity Assumption (HDCRA).

Since the Cloud-Paillier's variant is probabilistic and is secure under the HDCRA, the variant is semantically secure. To provide a high security level, the modulus n should be composed of two safe primes (i.e., Sophie primes [9]) and the primes do not fall in the capabilities of the fastest factorization attacks: the Number Field Sieve and the Elliptic Curve Method.

2.3.3.3. A Solution Based on Fully Homomorphic Encryption

The authors in [13] presented the solution as follows:

Encryption algorithm: The encryption parameters p, q and r, where p is a positive odd number, q is a large positive integer, p and q determined in the key generation phase, p is an encryption key, and r is a random number encrypted when selected. For the text m, the calculation is

$$n = m + 2r + pq \tag{2.2}$$

Then you can get the cipher text.

Decryption algorithm: to plaintext

$$m = (c \bmod p) \bmod 2 \tag{2.3}$$

Discussion: In this scheme users transmit and save their data to the cloud by encryption. This ensures the security of data in the process of transmission, and ensure safe storage of data. Although the cloud computing service providers handle the data, they can't easily obtain the information of the plaintext.

Fully homomorphic encryption mechanism enables users or the trusted third-party process cipher text data directly, instead of the original data. Fully homomorphic encryption

technology based on cipher text retrieval method can search directly on the cipher text data. It is not only to ensure query privacy and improve the efficiency of retrieval, the retrieval data can be added and multiply without changing the corresponding plaintext.

At present, the principal inconvenient that fully homomorphic encryption scheme has is a high computation problem that causes a very slow process between client and cloud provider. This problem is the subject of several studies.

2.3.3.4. Speedy Cloud-RSA

The authors suggest in [14], two variants of Cloud-RSA scheme in order to speed up its decryption process. The variants are based on modifying the form of Cloud-RSA modulus and exponents. The first variant (Rebalanced Cloud-RSA) uses a modulus formed of two or more distinct primes and employs the Chinese remainder theorem (CRT) to encrypt and decrypt. Whereas, the second variant (MultiPower Cloud-RSA) uses a modulus of the form:

$$n = p^r q^s, \tag{2.4}$$

for r ≥ 2 and s ≥ 1, And applies the Hensel lifting and the CRT to decrypt.

Discussion: The theoretical and simulation results in [1] show that the variants get a large decryption speedup over normal Cloud-RSA while maintaining a prescribed level of security. Rebalanced Cloud-RSA provides a faster decryption when using a small and secure modulus n < 2048-bit numbers. Whereas, MultiPower Cloud-RSA becomes more efficient with the increase in bit numbers of n.

2.3.3.5. Solution Based on Intel SGX

Securing cloud computing systems can be solved by hardware technology for protecting user data in cloud services at the lowest hardware level – into the CPU kernel. Starting with the sixth generation (Skylake), Software Guard Extensions (SGX) technology is implemented in Intel Core processors. This technology introduces a new mode of code execution in the processor, a memory protection mode, and a set of new instructions that ensure the SGX operation. Intel calls the SGX a reverse sandbox, i.e., instead of trying to isolate potentially malicious or untrusted software from the entire system, the developed program can isolate its code and its data from the entire system using the SGX technology. This technology is primarily intended for application developers who need to protect source code and data from unauthorized access and alterations, even if the malicious application is running at a higher privilege level. From the developer's point of view, the enclave represents an ordinary dynamic library (.dll and.so).

The main idea of this technology is to create a software enclave, a protected area of memory, access to which is controlled by the processor and is only allowed to the user application that created the enclave. The application developed using the Intel SGX

technology consists of two parts: unprotected and protected (enclave). On startup, the unprotected part creates an enclave in protected memory, transfers code and data from unprotected memory to the enclave and initializes it by performing integrity checks. Only protected functions see the data of the enclave in the clear. Any other external access, including by the operating system, is prohibited, due to total data and code encryption. Enclaves are areas in memory filled with encrypted pages of code and data. Enclaves are decrypted directly on the central processor when all checks are completed. This completely protects against the threat of information leakage in the open form to RAM or other I/O devices.

Discussion: In this solution, all mechanisms such as memory encryption, integrity control, and access control, are performed on the processor, which makes it possible to easily introduce Intel SGX to an already configured cloud system. The proposed approach to the organization of secure cloud computing based on the SGX technology ensures the confidentiality of data of cloud users at the processor level while preserving all advantages of cloud computing technology, which places the trust on the user's side on the manufacturer of Intel processors.

2.3.4. Confidentiality not by Encryption only

2.3.4.1. HPI-Secure

This solution [16] allows the user to store the encrypted version of the data in the cloud, without interrupting the functionality of the system. HPI_SECURE has five main components, namely an http analyser, an algorithm manager, a key manager, an exception manager, a distribution manager and an encryption manager. When there is a storage request on the server, HPI_SECURE calls the http analyser to read the information contained in the subject of the request (document content). The content is divided into different parts (two parts in this system). The algorithm manager randomly selects one of the standard encryption algorithms that the user prefers to use, such as RSA or AES 256. The key manager selects two different keys (for two parties) from a key group. If the user has created exceptions, the exception manager applies them to the data. The distribution manager manages the storage of the document after encryption. Finally, all of the above steps are combined and the details are stored in a database for later use. This also posts encrypted content to the server (s) according to the specifications of the distribution manager. When retrieving content, the same steps are followed by the encryption manager, but this time to decrypt the content and the display [17].

Discussion: The main target of HPISecure is the document-based applications. But there are still open challenges that the authors plan to cover in their future work, such as implementing HPISecure as a web-based application instead of desktop application (which violates the concept of mobility available in the cloud). Trading off availability for security is a critical point that is not deeply covered by HPISecure, they plan to include it in the next extension of HPISecure.

2.3.4.2. K-NN Classifier for Data Confidentiality

This approach classifies the data to be stored according to their security needs. This is done using the K-NN classification scheme [17] shown in Fig. 2.2. Data is classified into two classes, sensitive and non-sensitive. The RSA algorithm is applied to sensitive data while non-sensitive data is stored as it is. The K-NN machine learning technique is used to separate sensitive data from non-sensitive data. K-NN is used in a designed simulation environment. For reasons of precision, the value of k is kept at one (2.1). After separating sensitive data from non-sensitive data, the sensitive data is transmitted to the RSA algorithm for encryption. Non-sensitive data is directly assigned to a virtual machine without encryption.

Discussion: Many algorithms are used in text classification but K-NN is extensively discussed in the literature due to its simple implementation, high efficiency, better accuracy, and easily to understand. It is technically, a better approach to decide the security for the data based on the security level of the data instead imposing the encryption on complete dataset or just sending complete data to the cloud server without any proper security. The K-NN Classifier is a classifier that works well on basic recognition problems. Using a relatively large value of K may include some not so similar pixels. On the other hand, using a small value of K may exclude some potential candidate pixels. In both cases the classification accuracy will decrease. The computational complexity of K-NN depends on the size of the training dataset. If the training dataset is large, the computing complexity of K-NN is high. The two main steps of K-NN, which distinguish it from others, are distance/similarity measure and majority voting. One of the issues in K-NN is the approach to combining the class labels. The K-NN uses the simplest method to take a majority vote, but this can be a problem if the nearest neighbours vary widely in their distance and the closer neighbours more reliably indicate the class of the object.

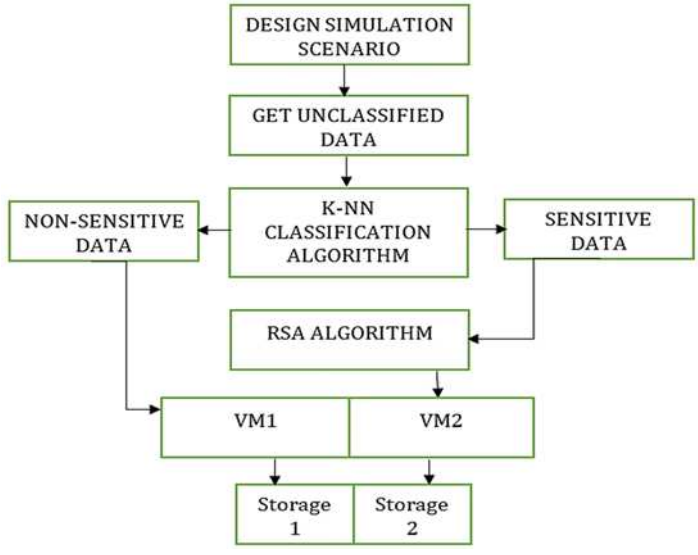


Fig. 2.2. K-NN classifier for data confidentiality [17].

2.3.4.3. Steganography-based Solution

In [18], the authors propose a new cloud computing scheme to improve the security and reliability of data. A secret image is encrypted using the Advanced Encryption Standard (AES) algorithm. Then, the encrypted image is incorporated into the host image via a steganography technique, which combines a discrete wavelet transformation (DWT) and a decomposition into singular values (SVD) to obtain the stego image. To preserve data integrity, a hash value is generated for the stego image using the secure hash algorithm 2 (SHA2) before the image is stored in the cloud. Once the image is extracted from the cloud, its hash value is generated using the same algorithm (i.e., SHA-2). The two hash values are then compared to check if the data stored in the cloud is changing and to obtain the secret image. The proposed scheme has been shown to be safe and highly effective through an in-depth analysis of safety and performance.

Discussion: Although existing systems have achieved a good level of data confidentiality, they still fail to maintain data integrity. Consequently, the authors of [18] therefore propose a secure system developed to obtain efficient performance while maintaining confidentiality with data integrity. They use first to ensure confidentiality, encryption based on the AES algorithm which they associate with a steganography technique which itself combines two other operations to obtain encrypted data called stego. Then to associate confidentiality with reliability of information, they use hashing using the secure hashing algorithm 2 (SHA2) before storing data in the cloud.

In [18] Section 5.3 assesses the robustness of the proposed method. Their system exhibits higher anti-interference performance and stability than pure SVDs against various malicious attacks. Table 2.4 shows the efficiency in terms of calculation time for incorporation and extraction (in seconds).

2.3.4.4. Confidentiality Preserving Through Encryption and Obfuscation

Normally, privacy is achieved by encryption. But encryption alone cannot provide security. In [19] the authors used both encryption and obfuscation to maintain confidentiality (illustrated in Fig. 2.3). Like encryption, obfuscation must work in association as reverse engineering attacks or brute force techniques can break this. Obfuscation is built into the encryption. Obfuscation uses a mathematical function or uses programming techniques to conceal illegal users. This approach depends on the type of data. Encryption can be applied to alphabets and alphanumeric data types and obfuscation to numeric data.

There are 3 different algorithms present in a model involving encryption and obfuscation:

- Algorithm #1 is used to find out the type of data (T) which is ready to store in the cloud storage;
- Algorithm #2 is used for obfuscation;
- Algorithm #3 is used for encryption.

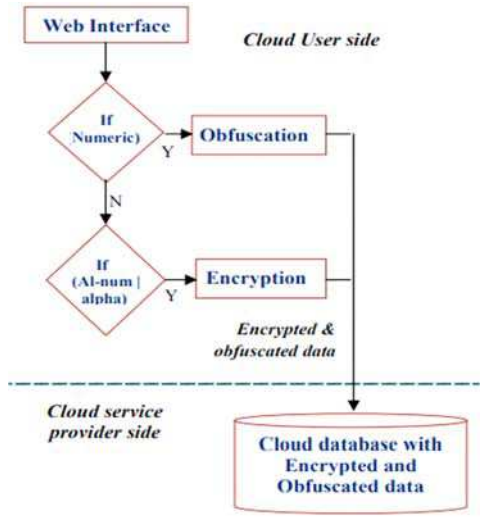


Fig. 2.3. Confidentiality through encryption and obfuscation [19].

Discussion: Proposed cryptography technique uses encryption and obfuscation for the different type of data. Integration of obfuscation technique with encryption technique has given more confidentiality than they used in separately. Confidentiality of cloud data is ensured by using this technique. This can protect the data in cloud storage from insiders as well as outsiders' attack.

Encryption only or obfuscation only is not sufficient for cloud data storage. Integration of both techniques should provide maximum security to user's data in the cloud data.

2.3.4.5. Attribute-based Proxy Re-encryption

To ensure data confidentiality and fine-grained access control in cloud computing environments, in [11] there was proposed a system model using Key Policy-Attribute Based Encryption (KP-ABE) and Proxy Re-Encryption (PRE). Formally, [11] ensures data confidentiality using KP-ABE and sending the data owner delegate computation overload to the proxy using PRE. Also, [11] introduces user revocations into the system model. But the proposed scheme is vulnerable to collusion attack using a revoked user and a cloud server. In addition, there is no selective delegation of the level of trust, i.e., revealing partial user information to A agency, but not to B agency. To resolve these challenges, the authors of [20] divide data files stored on cloud servers into a header and a body as in [11]. Decryption rights such as head (encrypted key using KPABE) and body (encrypted message using symmetric encryption) are allocated cloud servers as in [11]. For this reason, cloud servers become too powerful but cannot be trusted. Therefore, they create a trusted authority named the privilege manager to manage the header while storing the body on the cloud server. Thus, they split the power that was concentrated in the cloud servers. Finally, they introduce a data access privilege management model using Type-based Proxy Re-encryption's concept for mobile cloud environments.

To be more precise, data file stored in the cloud server is divided into header and body to store to the privilege manager group and the cloud service provider. In the header, DEK used for encrypting data into body is encrypted to KP-ABE. In other words, if the privilege manager group does not authorize the user, he cannot gain information of DEK within header, and actually, the user intends to gain data within body becomes completely impossible. Thus, data confidentiality is guaranteed by storing and dividing data file stored in cloud server into header and body. Also, by having privilege manager group within cloud, user access is managed, and selectively delegation according to weight (type information) set by the data owner is enabled. To specify such function, Type-based PRE is utilized. According to the level the data owner trusts the user group, the data is divided into the whole part or parts, and type information is granted to each, and is encrypted. Through this, data owner can selectively delegate decryption right for decrypting the whole or part of the data to the user group.

Discussion: The paper [11] suggests a model which can guarantee data confidentiality and fine-grained access control. However, as mentioned above, it is vulnerable to the collusion attack of revoked user and cloud service provider. It is because authority related to decrypting such as encrypted DEK, encrypted data (header and body, etc.) is concentrated to the cloud service provider. The proposed scheme in [16] manages key (header) by having trusted authority called privilege manager group, and the cloud server stores only the encrypted data (body), That is, proposed scheme prevents collusion attack and guarantees data confidentiality. Additionally, by utilizing Type-based PRE, weight is granted according to the data owner's reliability towards the user to show the whole or part of the data. That is, selectively delegating message is possible.

2.3.4.6. TR-MABE

The authors in [21] claim that existing work on secure and privacy preserving access control in e-healthcare systems cannot achieve the security and privacy requirement for cloud computing. They proposed thus a solution called TR-MABE to realize all the security and privacy requirements with significantly enhanced efficiency. Their main contributions are outlined as follows:

- A white-box traceable and revocable multi-authority attribute-based encryption TR-MABE is proposed for fine-grained multiple level access control in e-healthcare cloud computing systems;
- The property of traceability is realized to protect the patient’s private PHI from being abused and exposed for target advertisement, since once the secret key is illegally leaked, the source of PHI will be precisely traced;
- Without introducing extra special signatures, the multilevel privacy preservation is efficiently realized by outsourcing the storage revocation (i.e., cipher text updating) to the e-Healthcare cloud.

Discussion: In the proposed TR-MABE, multilevel privacy preservation is achieved by exploiting the technique of revocable storage in the multi-authority setting. Without an

online TA, the proposed construction can simultaneously trace the physicians who have leaked the secret key to unauthorized entities for potential patient's PHI and identity exposure. This functionality can also not be achieved by group signatures.

The authors illustrate the computational cost comparison among Boyen's scheme [22], Liang's scheme [23] and their proposed TR-MABE respectively on the patient and physician's ends. It is observed that the computational cost increases as the size of physician's attribute set Natt grows. However, by exploiting the technique of group signatures [23, 24] widely adopted to achieve conditional identity privacy, the computational cost increase as the number of physicians grows since it is required for the patient to generate one group signature for each physician in the PKI setting. Significantly from the existing techniques [23, 24], the proposed TR-MABE requires no extra special kind of signature to achieve multilevel patient identity privacy and naturally embraces this functionality by designing the technique of traceable and revocable multi-authority CPABE.

In Fig. 2.4 [21], it is obviously observed that the communication cost of Boyen's scheme [22] and Liang's scheme [23] sharply grows as the number of physicians increases from 50 to 500 in the healthcare provider and the size of physician's attribute set Natt increases from 10 to 30. However, the communication cost of the proposed TR-MABE is significantly lower than [23, 24] and independent of the physician number.

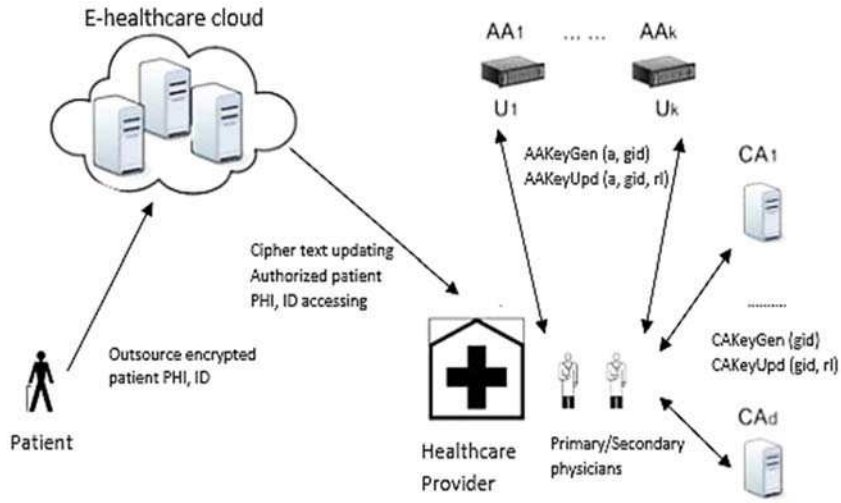


Fig. 2.4. Architecture of e-healthcare cloud computing system [21].

2.3.4.7. Privacy-manager

In [24], the authors describe a privacy manager for cloud computing, which reduces the risk to the cloud computing user of their private data being stolen or misused, and also assists the cloud computing provider to conform to privacy law. This proposed solution helps the user manage the privacy of their data in the cloud.

As a first line of defense, the privacy manager uses a feature called obfuscation, where this is possible. The idea is that instead of being present unencrypted in the cloud, the user's private data is sent to the cloud in an encrypted form, and the processing is done on the encrypted data. The privacy manager to reveal the correct result de-obfuscates the output of the processing. (We call it obfuscation rather than encryption because some of the information present in the original data is in general still present in the obfuscated data.).

Different possible architectures for privacy management within cloud computing are listed:

- Privacy manager in the client;
- Privacy manager in a hybrid cloud;
- Privacy infomediary within the cloud.

Discussion: The authors do not expect obfuscation to protect data from a very determined adversary. In particular, some obfuscation methods are highly susceptible to known plaintext attacks. However, obfuscating private data does at least protect the data from opportunistic data thieves with access to cloud databases, because it ensures that the data is never present in the database in the clear. This solution is not suitable for all cloud applications.

2.3.4.8. Hierarchical Classes of Computing for Privacy Preserving

The authors in [25] argue that cryptography alone cannot enforce the privacy demanded by common cloud computing services, even with such powerful tools as FHE (Fully Homomorphic Encryption). They treated in their document, a cloud for simplicity as a highly resourced, monolithic entity called S. The overarching goal of privacy preservation they explore dictates that in no case should S learn any portion of any piece xi of private data.

They explore the challenge of privacy preservation for clients in the cloud by proposing a (nested) hierarchy of three classes of privacy-preserving forms of computing. These classes are meant to characterize natural applications that S might be called upon to execute in the cloud over clients' sensitive data. They define privacy preservation here to mean that S itself should learn no information from any application execution, while select clients should learn limited output information. The three classes, in order of increasing generality, are:

- Private single-client computing;
- Private multi-client computing;
- Stateful private multi-client computing.

Discussion: The authors have shown the limitations of cryptography alone in meeting the challenges of cloud privacy. In their scheme, and more precisely in the second class which

is private multi-client computing they proved that it cannot be realized with cryptography alone, thus such applications require trusted state / execution of some sort. Since they claim that cryptography alone is not enough, they propose several practices for reliable calculations among which we note trusted computing.

While as always helping demarcate trust boundaries, cryptography will also help verify specific security requirements of cloud deployments, e.g., correctly configured storage. It will be one supporting component in a complex ecosystem of trust that depends on interlocking technical, regulatory, and commercial security and privacy enforcement approaches.

2.3.4.9. Encryption and Trust-based Solution

One of the biggest challenges in the cloud is the efficiency with which cloud service providers (CSPs) can protect the privacy of data for cloud service users (CSUs). This approach [26] (illustrated in Fig. 2.5) aims to guarantee confidentiality by allowing CSUs:

- To encrypt sensitive data and check their accuracy from time to time;
- To assess the reliability of the CSP;
- To decide whether the CSPs can run various calculation services according to their confidence values.

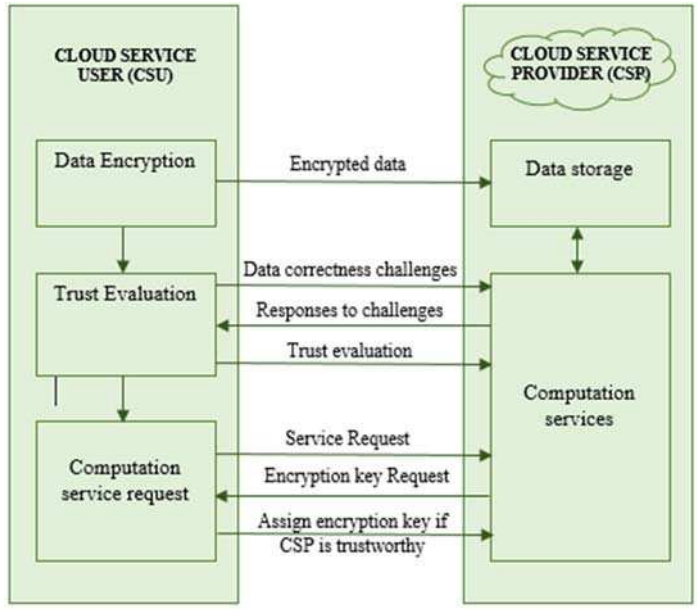


Fig. 2.5. System model of the encryption and trust-based solution [26].

In most encryption techniques, the big question is who should manage the encryption keys. Trust becomes a critical issue if a third party manages the key. If it is with CSP,

again, reliability appears, because CSPs are not always trustworthy. This method follows the strategy that CSUs manage the encryption key and thus have more control over their own data. But this has a certain drawback. CSPs cannot provide certain services such as data search, changes, additions, deletions and keyless insertions. Therefore, CSUs must decide to assign encryption keys to a given cryptographic service provider. Determining whether to assign keys to cryptographic service providers is a critical issue. This requires a method that can help key holders assess the risks associated with assigning keys to third parties. This approach called EnTrust attempts to integrate encryption and trust techniques to improve privacy in cloud computing.

Discussion: The proposed framework contains three components: an encryption module, a trust evaluation module and a decision module. Through the encryption module, CSUs are able to encrypt their sensitive data before storing it in the cloud. Furthermore, the encryption schemes selected in this work allow CSUs to verify the data correctness from time to time and enable searches over the encrypted data by CSPs with provided encryption keys. However, to utilize the computation services provided by CSPs, CSUs have to expose some private information to CSPs. Therefore, CSUs have to carefully select trustworthy CSPs to perform the computation services. The trust evaluation module provides an effective way for CSUs to calculate the trust values of CSPs based on CSPs' behaviours. With such trust values, the CSUs are able to make decisions about whether to allow a CSP to perform a certain type of data computation service.

To ensure a high-level security, in this paper, the authors propose that CSUs perform both the encryption and trust evaluation process, which may cause relatively heavy workloads for CSUs, especially individual cloud users. In a future work, they consider to involve third party auditors who can take over complicated computations and reduce CSUs' workload. Then the question is: how CSUs can assign workloads to the third-party auditors while not exposing too much sensitive information?

2.3.4.10. Blockchain for Cloud-based Data Privacy

The authors in [27] studied the potential to use the blockchain technology to protect healthcare data hosted within the cloud.

There has been recent interest in utilizing blockchain in the provision of secure healthcare data management. Broadly speaking, blockchain is a technology able to build an open and distributed online database, which consists a list of data structures (also known as blocks) that are linked with each other. These blocks are distributed among multiple nodes of an infrastructure, and are not centrally stored. Each block contains a timestamp of its production, the hash of the previous block and the transaction data, and in this context, a patient's healthcare data and the healthcare provider information. Fig. 2.6 describes the authors conceptual blockchain-based EMR (Electronic Medical Records)/HER (Electronic Health Records)/PHR (Personal Health Records) ecosystem. Specifically, when new healthcare data for a particular patient is created a new block is instantiated and distributed to all peers in the patient network. After a majority of the peers have approved the new block, the system will insert it in the chain. This allows them to achieve a global

view of the patient's medical history in an efficient, verifiable, and permanent way. If the agreement is not reached, then a fork in the chain is created and the block is defined as an orphan and does not belong to the main chain. Once the block has been inserted into the chain, the data in any given block cannot be modified without modifying all subsequent blocks.

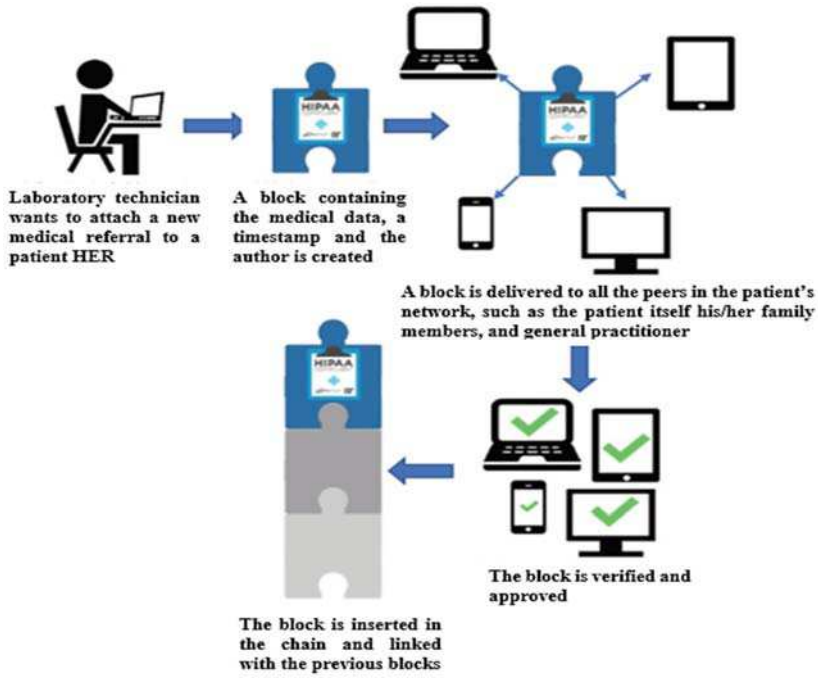


Fig. 2.6. A conceptual blockchain-based EMR/HER/PHR ecosystem [27].

Discussion: securing EMR/EHR/PHR ecosystem and ensuring privacy and integrity of the data is an active research area. Many approaches include using cryptographic primitives, such as those based on public key infrastructure and public clouds. However, this limits the searchability of the data, in the sense that healthcare providers have to decrypt the data prior to searching on the decrypted data, resulting in increases in time and costs for the data retrieval and diagnosis.

The strong data integrity feature of blockchain results in immutability that any data, once stored in blockchain, cannot be altered or deleted. However, if the record were healthcare data, then such personal data would come under the protection of privacy laws, many of them would not allow personal data to be kept perpetually.

Another practical issue is on how fit it is for blockchain to store healthcare data. Blockchain was originally designed to record transaction data, which is relatively small and linear. Healthcare data, such as imaging and treatment plans, however, can be large and relational that requires searching. How well blockchain storage can cope with both requirements is currently unclear.

In order to deal with these challenges, many have suggested the notion of off-chain storage of data, where data is kept outside of blockchain in a conventional or a distributed database, but the hashes of the data are stored in the blockchain. This is said to be the best of both worlds, as healthcare data is stored off-chain and may be secured, corrected, and erased as appropriate. At the same time, immutable hashes of the healthcare data are stored on-chain for checking the authenticity and accuracy of the off-chain medical records.

2.3.4.11. L-EncDB

For the protection of data privacy, sensitive data usually have to be encrypted before outsourcing, which makes effective database utilization a very challenging task. To address this challenge, in [28], the authors proposed L-EncDB, a novel lightweight encryption mechanism for database, which keeps the database structure, and supports efficient SQL-based queries. To achieve this goal, a new format-preserving encryption (FPE) scheme is constructed, which can be used to encrypt all types of character strings stored in database.

In the proposed L-EncDB system, shown in Fig. 2.7, the encryption and query are based on SQL. Through only one interface, all SQL sentences for database can be interpreted. Furthermore, based on format-preserving encryption (FPE) technique and a new character string FPE scheme, L-EncDB can be implemented to preserve data type and length in cipher text.

Discussion: L-EncDB is an external encryption mechanism independent from database. The adopted encryption algorithms affect DB operation performance. To evaluate its performance, two issues are addressed:

- The performance of FPE encryption algorithms for batch data encryption;
- The performance of OPE.

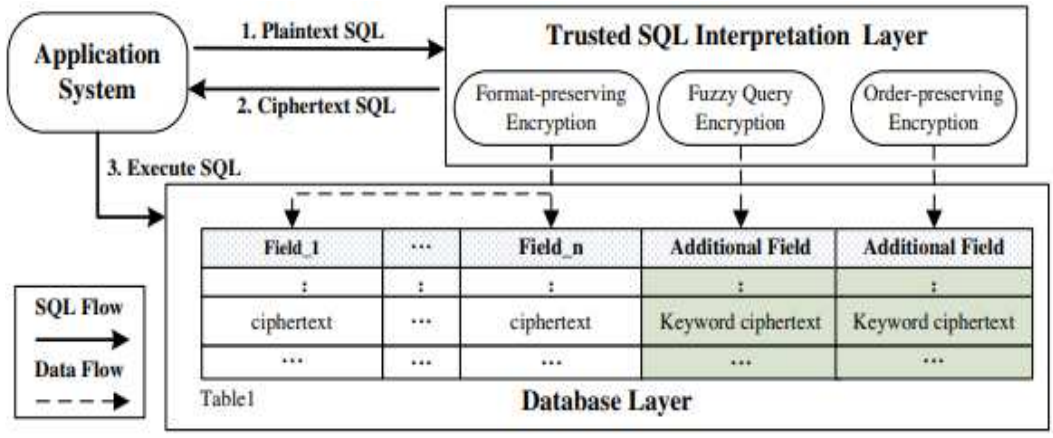


Fig. 2.7. L-EncDB architecture [28].

2.3.4.12. CS-IBE

The authors in [29] proposed a scheme based on Identity-Based Encryption (IBE) through Pairing Based Cryptography (PBC) library, in order to ensure data confidentiality against accidental, malicious, and illegal access. The goal is to encrypt data before outsourcing them. The design enables cloud users to store and manage their data securely. It also ensures secure, adaptable and extensible key management inherent to IBE without a growing key store.

This is how CS-IBE scheme works: Each file is associated with at least one policy P (owner's ID) and other attributes can be added if needed (e.g., DateTime expiration). The file is encrypted with the data key (generated by the cloud user using symmetric encryption AES-128 using CBC mode of operation), and the data key is further encrypted by the control key generated through the IBE mechanism. CS-IBE acts as an overlay system that works atop cloud storage solutions and do not require any engineering changes on the cloud storage to be supported. The components of the implemented design are: the cloud User, the KM and the cloud Storage. At a first stage, the cloud user needs to get the mpk (master public key) from the KM for cryptographic operations. Once downloaded it can be stored for future use. For each File F, the cloud user generates a symmetric key K to encrypt and to generate the ciphered dataCF, and then encrypts the data key by IBE mechanism CK.

Discussion: Compared to other proposed designs, CS-IBE is dedicated, at a first stage, to provide confidentiality of outsourced data and to make them unreadable for other entities. Our design is slightly different in three aspects: First, CSIBE ensures double protection of outsourced data by adding the security layer on data key K that is stored encrypted. Second, CS-IBE enables the data access anywhere and at any time, in case that cloud user is outside the KM perimeter, he can ask for his corresponding private key dP. Hence, it is up to him to ensure its privacy, this can be done using the Trusted Platform Module (TPM) since most released computer have now the chip integrated in the mother board. The Third advantage is that CS-IBE is lightweight and user friendly. Cloud user does not have to worry about the management of cryptographic keys and have not to worry about a growing keys and certificates store. The only issue is the security parameter P used by the KM to generate the whole system parameter in the setup function.

In addition, the design enables confidentiality protection for replicated data. When an attacker gets access to storage intentionally or accidentally, he will not be able to derive sensitive information since they are encrypted.

2.3.4.13. GRBAC

The concept of GRBAC began with multi-applications and multi-users system, the central idea of this scheme is that permissions are associated with the assigned roles and users are assigned to these roles based on their context information. GRBAC is the secure RBAC server that contains the roles which are assigned obliviously to user, keeping the user identity anonymous, permissions and constraints associated with the users are also

encrypted. Roles are assigned for various job functions within the organization. This model consists of two units: one creating the garbled policies and obliviously assigning roles to the users and other performing verification of users and secure evaluation of the policies. The GRBAC server is for performing computations on garbled (Encrypted) data. Services are being provided by the GRBAC server who assigns the roles obliviously and encrypt the content (permissions, constraints, policies) before placing them on Cloud. Cloud or policy evaluator is the third party that takes garbled data and performs computation on it. Users provide their encrypted attributes for the evaluation purpose, who want to access the resource based on the roles assigned to them. Policy published on GAC are hidden from it due to its garbled characteristics, which helps to maintain the privacy of roles assigned to the users and their private attributes, in case GAC gets compromised.

Discussion: This proposed model incorporates the garbled layer to provide efficient security to the cloud access controls. It has key management mechanism of sharing keys between the GRBAC server and the user. User has its own secret key i.e., only user knows the permissions associated with his assigned roles. GRBAC is well suited for secret organizations where identity of the user has to remain secret even to the organization itself.

This solution compared to all existing scheme keeps the user's ID secret, the roles used are unknown to the cloud and the strategies remain garbled (see Table 2 in [30]). In addition, during authentication, each user is provided with a token/S_{ID} pair and the calculation is performed on encrypted data. Finally, with regard to security and flexibility, this solution is judged from Table 2 in [30], very efficient.

2.3.4.14. ISRSAC

The indirect way of protecting the files against attackers is by increasing the confidentiality through data encryption. The proposed ISRSAC public key cryptosystem [31] focuses on increasing the time to perform factorization and randomization. Many researchers' variants of RSA security depend on the value of 'n'. Since the value of 'n' is the multiplication of two prime numbers 'p' and 'q', it can be factored using integer factorization methods. Therefore, the enhancement of RSA needs to be relied on the difficulty in factorizing the value of 'n' and randomness. All cloud users generate the public key and the private key. The proposed ISRSAC key generation uses two prime numbers as same as RSA but, the values of 'p' and 'q' must be greater than 3. The public key 'e' and private key 'd' depends on the public key 'n', which is the multiple of 'p', 'q', 'p – 1' and 'q – 1'. In order to increase the security, randomization is essential in a cryptosystem. The random number 'r' is chosen for 2^r, which must be between the values of 'p' and 'q'. Instead of applying Euler's totient function, secure function $\alpha(n)$ is calculated based on the random number.

Discussion: According to the authors in [31], ISRSAC is at least as hard as factoring n and finding randomized number r, obtaining private key 'd' from the public encryption key 'e' and computing D(n) without factoring n.

Based on the performance analysis of ISRSAC, the security of ISRSAC is high, compared with RSA cryptosystem. In RSA, the value of 'n' is based on only two prime numbers. However, in ISRSAC, the value of 'n' depends on four numbers, where two numbers are prime. In RSA, Euler's totient function Φ(n), public key 'e' and private key 'd' depends on the value of 'n'. And in ISRSAC, secure function α(n), public key 'e' and private key 'd' depends on the value of 'n'. So, the key generation, encryption and decryption time are slightly higher than RSA, which is tolerable. In cloud, for efficient and secure data storage and data sharing, ISRSAC can be used to improve the data confidentiality.

2.3.4.15. Solution Using Modular Encryption Standard

The proposed scheme comprises three major steps: Identification, Classification, and Securing:

- **Identification:** the need for protecting nuclear information is regulated by performing identification and classification according to the degree of risk factor. Here, the authors have to identify the sensitivity and criticality of the data. The identification of nuclear information is based upon user specified requirements. It generally has two broad categories: confidential information, which requires protection, and public information which does not require protection.
- **Classification:** Nuclear information classification scheme decides the level of confidentiality of the information. This is helpful in choosing the information that actually should be protected, and it ultimately diminishes the cost of security. The authors have classified these two categories into five different subcategories (based upon the sensitivity level): non confidential Information, confidential information, less confidential information, moderately confidential information, highly confidential information, extremely high confidential information.
- **Securing:** there are several sub-parts in the security procedure which explains the strength of this scheme: Extension/Contraction, Key Whitening, encryption, decryption, key transformation, Key Encryption and Decryption.

Discussion: This solution has the following advantages:

- It provides 5 keys (two extra keys than AES), but with no extra bit utilization (256 bits same as AES);
- It constitutes key encryption as well (to enhance security);
- It is a modular algorithm. Some modules are at the user side (autogenerated), though extender/contractor module data will be extended before uploading to any cloud and other modules for the intermediary cloud. Therefore, the overall benefit is that data will not be handed over in actual form to any third party;
- Every single key is treated in two different ways, i.e., half key for XORing and half key for XNORing (except key whitening);

– Every single key transforms the data two times (except key whitening). Therefore (except key whitening), the key intermixes 18 times with data instead of 9 times (for 9 rounds), as key addition and key subtraction are, in actual, the data and key subsuming. This is the major reason behind 64-bit data and using a large size key to subsume the single block of data (8×8 matrix) by a single key twice.

This technique is proposed to protect against insider and outsider's attack due to multilevel security and multicloud utilization.

2.3.4.16. CPDA

The authors in [33] have proposed a confidentiality-preserving deduplication cloud storage with public cloud auditing which:

– Simultaneously supports secure file deduplication on encrypted files and authentication tags deduplication. This scheme also provides the public integrity verification of the unique file in deduplication cloud storage by using cloud auditing technology;

– Realizes the deduplication on encrypted files by introducing the convergent encryption technique which guarantees the same plaintext file can produce the same ciphertext. Thus, the CSP (Cloud Service Provider) and TPA (Third Party Auditor) cannot obtain any actual content of DOs' file during the process of deduplication and public auditing and the confidentiality of the outsourced data is guaranteed;

– Not only supports each Data Owner (DO) to independently delegate the integrity auditing of their own files to the TPA, but also supports CSP to periodically delegate TPA to concurrently handle multiple auditing tasks to ensure the integrity of outsourced files.

Discussion: The authors have shown in [33] that the solution is comprehensively efficient and can support all the given functions in a more effective way. The security of the CPDA scheme also is formally proven. The security model of CPDA scheme is defined from the perspectives of integrity auditing and POW. First, CPDA scheme is required to be CSP-unforgeable in the integrity auditing process according to the scheme in [35]. Second, CPDA scheme is required to be (k, θ)-uncheatable in the POW process according to the security definition in [34].

CSP-unforgeable: A CPDA scheme is CSP-unforgeable if no dishonest CSP can successfully forge the auditing proof to deceive the TPA with a non-negligible probability.

(k, θ)-uncheatable: A CPDA scheme is (k, θ)-uncheatable if given a file F with min-entropy k, no malicious user, who knows partial file containing θ-bit Shannon entropy of F, convinces the CSP that he has the entire file F with a non-negligible probability.

2.3.5. Confidentiality According to the Type of Storage

2.3.5.1. Secret Sharing Scheme

In [36], the proposed scheme is used with a private / public hybrid cloud storage approach. This is achieved by developing a secure and fully automated data storage and transfer protocol in the cloud between cloud storage providers and consumers. In this approach, sensitive data are secured in the private cloud infrastructure before being sent to a public cloud for storage. This protocol is implemented in two phases with a secret sharing scheme. The send phase occurs when a file is being stored or updated in the public cloud storage and the retrieve phase occurs when a file is being accessed or downloaded from the cloud infrastructure. The file send strategy is as follows: Having a public cloud storage provider and 'n' available virtual volumes V1, V2,...,Vn, with each having certain storage space. When a file is to be uploaded into the public cloud, it is split into many smaller chunks. These chunks are then stored in the available volumes in such a way that it becomes impossible for the attacker to reconstruct the initial file. Every chunk upload operation is stored in the chunk distribution dictionary. Every record in this dictionary has three vital information:

- Index of the chunk in the original file which provides the exact position in the file;
- Virtual volume(s) where the chunk is stored;
- Hash sum of the chunk that is used for integrity check.

This dictionary is stored on the private cloud infrastructure. The process of splitting the file into chunks also happens in the private side and the chunks are sent to the public cloud in an order different from the one obtained from the source file.

Discussion: The strong point of this method is that the secret can be reconstructed only when a predefined number of k shares are combined together; individual shares are of no use on their own, so anyone with fewer than k out of n shares has no extra information about the secret than someone with 0 shares. The size of each share does not exceed the size of the secret. Keeping k fixed, shares can be easily added or removed, without affecting other shares. It is easy to change the shares, keeping the same secret. Shamir's scheme is a perfect secret sharing scheme, as k-1 shares are just as useless as no shares. Each share is p-bits long. The value of p should not be too small or it could be susceptible to brute-force attack. If the value of p is 128 bits, then this provides 2128 possible values, which is a range too large for brute force to ever attempt.

2.3.5.2. Scheme for Image Data Privacy

The traditional method to protect the privacy of data is to employ cryptographic algorithms, which unavoidably introduces heavy computation. Another way is hybrid cloud consisting of public and private cloud. The sensitive data is separated from non-sensitive data, and only the non-sensitive data is outsourced to public cloud. If we use hybrid cloud method directly, all the private images have to be stored in private cloud,

which makes the adoption of cloud computing meaningless. Besides achieving data privacy, we should reduce computation and storage overhead in private cloud, as well as communication overhead between private and public cloud. In [37] the authors propose a novel scheme to achieve the above goals.

The architecture of a hybrid cloud is illustrated in Fig. 2.2. The original data come from the private cloud and are sent to a data processing centre. For image data containing sensitive information, the authors suggest to divide the image into several blocks with the same size, and add random noise into each colour dimension of every block, which breaks the pairwise blocks relationship. Then they shuffle these blocks with a random permutation, which makes the image becomes unreadable, and send it to the public cloud. They store the random noise and the information used to shuffle blocks in the private cloud. The ID of the image is sent to the private cloud and public cloud at the same time. The shuffle order is obtained from the private cloud to reorder the shuffled image downloaded from the public cloud. Then the random noise is obtained from the private cloud to recover the original image. They consider an un-trusted public cloud who intends to get sensitive user data and has full control of the hardware, software, and networks in the public cloud.

Since the shuffled and modified image containing sensitive information has become unreadable, it can be put in public cloud. As shown in Fig. 2.2, the key information used to recover data is transmitted between data centre and web server within the private cloud, a user is not able to obtain the key information.

Discussion: To evaluate the efficiency of their privacy preserving method, the authors compare their algorithm with the Advanced Encryption Standard (AES) algorithm, which is a standard cryptographic algorithm based on permutations and substitutions. For each size of image, they run 100 times modification and recovery operations and get the average time. The results show that the time for processing image increases when the image size becomes larger. For all the block sizes, the algorithm is about 1,000 times faster than the AES algorithm. The reason is that the algorithm has no iteration, while AES consists of four stages with many rounds. To sum up, the algorithm provides image data privacy and it is much more efficient than AES.

The authors implemented the scheme in real network environments (including the Amazon EC2) and tested the security, efficiency and communication overhead. Both the analysis and experimental results showed that this scheme is secure, efficient and has little overhead.

2.3.5.3. DCSSP

In [38], Alexandru Butoi and al. assess the security strength of their DCCSP protocol using a brute force attack. They consider two interacting parties: the consumer and the cloud provider. The consumer can be any user who contracts and uses cloud storage resources to store their data. The cloud provider is an organization that owns, administers and leases cloud resources to the cloud consumer. This protocol illustrated in Fig. 2.9,

ensures that before storing data in the public cloud, it will be secure by calculating the secret sharing scheme in the user's private cloud. The secret is the file that contains sensitive information and the sharers are public cloud storage volumes. As a result, only the consumer in the cloud is aware of the data distribution model in the public cloud. The protocol defines two main phases: sending the file and retrieving the file.

Discussion: Many of the secret methods of sharing data protection in the cloud [3, 4] use multiple cloud providers, while DCCSP uses multiple virtual storage systems from a single provider.

We have found that the probability of unauthorized data reconstruction can be reduced exponentially by using a sufficiently large number of chunks. We also observed that the increase in the number of storage volumes used resulted in a linear decrease in the reconstruction of unauthorized data. This suggests that DCCSP can be profitable by using a constant number of storage volumes (fixed cost) and by exploiting the number of blocks to achieve high levels of security. Regarding future work, it would be interesting to develop a new variant of the word-level separation algorithm for text files.

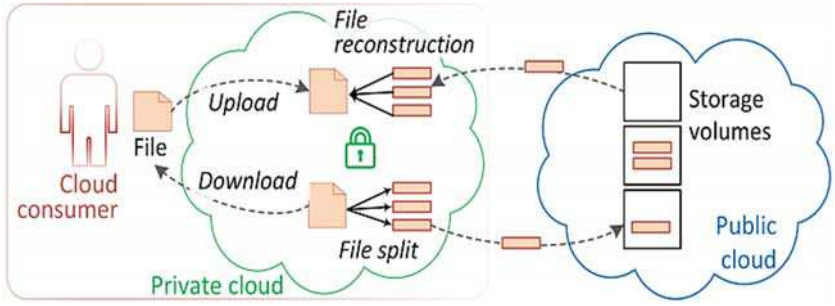


Fig. 2.9. Protocol DCSSP overview [38].

2.3.5.4. CAFE

To support confidential execution of various kinds of applications in a practical cloud marketplace setting, the confidentiality of sensitive logic should be systematically maintained in the entire workflow from the development of programs to the delivery to the cloud users and their execution. CAFE achieves the aforementioned goals using an end-to-end framework for the confidential execution of cloud applications by using hypervisor-level techniques. This is one of the key novelties of this scheme compared to previous work which only focus on a local view of protection. CAFE works in the following way. First, developers create software or port existing software in two groups of program binary files: a group that can be open to cloud users and the other that contains confidential logic that needs to be protected. They name the binary files of the former public binaries and the latter secret binaries. The public binaries are submitted in the form of a VM image that also contains other files related to the application and the VM environment but does not contain the secret binaries. A cloud user may be able to copy the files or extract the in-memory images of the public binaries as in the existing cloud

application distribution. In contrast, the secret binaries are submitted in the form of separate files and are automatically managed by the cloud providers with protection. When the application is run in the user VM, the secret binaries are fetched on demand at runtime for execution via a secure deployment protocol by the hypervisor. The hypervisor securely loads the secret binaries through a cryptographically protected channel after the authentication of the VM. The end-to-end framework ensures that the sensitive logic is completely isolated from cloud users at all times. Throughout the whole lifetime of the VM, the binary and runtime states of the sensitive logic stay confidential and are strictly protected from the entire guest OS in the VM by the hypervisor.

Discussion: In [39], the authors have shown:

- The applicability of CAFE to various types of application software to verify that similar program logic can be executed confidentially;
- That various types of application code can be protected by CAFE using 9 applications grouped into three distinct categories (Decision-making logic, cryptographic operations, Data processing workload).

2.3.5.5. ARIANNA

The authors in [40] propose a mobile secure storage approach which is primarily oriented to guarantee confidentiality and integrity issues concerning data stored on mobile smart devices that are part of a multi-Cloud environment. To this end, they introduce and discuss the "ARIANNA" app. ARIANNA consists of an Android app which is totally compliant and integrated in the SSME Cloud Service [41], a novel experimental framework of secure storage service in multi-Cloud environment which is deployed and maintained at the Cloud Data Center - University of Messina.

Discussion: ARIANNA addresses confidentiality and integrity issues concerning data stored in mobile devices with the aim to protect them against insider attacks.

2.3.6. Confidentiality Without Encryption

2.3.6.1. NetDB2-MS

The authors propose, in [42], a new model applied for NetDB2 architecture, called NetDB2 Multi-Shares (NetDB2-MS). NetDB2-MS ensures privacy in DAAS and is based on multi-service providers and the secret sharing algorithm instead of encryption, which was used by the existing NetDB2 service. The purpose of the proposed new model is to avoid the high cost of data encryption and decryption.

Fig. 2.10 illustrates the general DAAS model. Part A represents the client side, which sends data inquiries to the server in part B. The data source in part C stores the data on the server side. The server side, which is supposed to be a trusted server, ensures the privacy of any client query. A problem occurs if we cannot guarantee that the server is a trusted

service. In a lot of research that has use encryption techniques to protect data from an un-trusted server, the encryption and the decryption process takes place at the data source in part C. The proposed NetDB2-MS model does not preserve privacy by encryption; rather, privacy is preserved by using the secret sharing method. This avoids the negative effects of encryption on queries. It focuses on the procedures between the data source and the service providers, not between the client and the http server.

Discussion: the authors compare data storing time between the secret sharing algorithm in NetDB2-MS and the blowfish algorithm in the NetDB2 service with various data sizes in several cycles. They show that the blowfish algorithm outperforms the secret sharing algorithm when they increase the data size. Although blowfish algorithm is faster than secret sharing algorithm, but typically the data will be distributed once to multiple shares. This does not affect the contribution for ensuring the privacy of users' queries during the data retrieval procedure.

It was found that the number of solutions for data privacy in DAAS using encryption techniques were higher than those for any other techniques. The use of multi-sharing techniques is considered novel. It is proven to be superior to the encryption technique due to the latter's limitations and its negative impacts such as high time cost for encryption and decryption of the data.

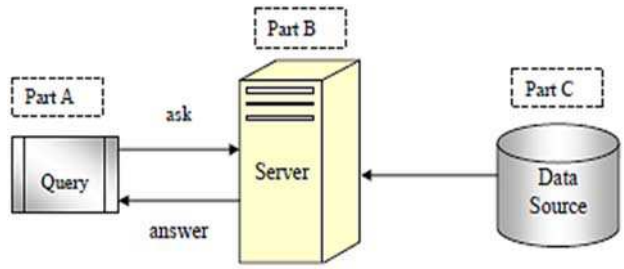


Fig. 2.10. Client/Server DAAS model [42].

2.3.6.2. Multiple Storage Solution

In [43], the authors discuss the challenges, advantages and disadvantages of existing solutions for data security and confidentiality in public cloud computing.

One method of protecting data privacy is distributing files across multiple stores. Multiple storages are applied to minimize information leakage when a single storage is compromised. With this method, encryption is not necessary.

[36] Proposes optimal algorithms for splitting and distributing data in order to minimize the useful informative content contained in each block of file stored in different virtual memories. For each file fragment ci and for the whole file f, the objective of the split is to obtain the largest relative entropy I (f, ci), which corresponds to the information lost when ci is used to approach f. To rebuild f, you need to find the correct set of virtual memories

among all the virtual storage volumes, as well as the correct sequence of pieces of files among all the pieces of files in this set of virtual storage volumes. For an insider attack, we know the storage game, but not the sequence. Distributing files is intended to minimize the likelihood of successful recovery.

Discussion: The proposed architecture could provide various security methods depending on the type of data and the use in order to reduce complexity by calling a common API. However, this design poses some problems:

- The call to the common API is intensive, caching and scalable procedures are necessary;
- Although most storage and compute operations are performed in a remote cloud, pre-processing and some data information must be stored and kept locally;
- Since algorithms using encrypted data are applied, client-oriented algorithms must be developed. Users must have a good knowledge of encrypted data;
- A fully functional reference for the system must be designed, implemented and evaluated with quantitative and qualitative performance measures.

2.3.6.3. MONcrypt

The proposed technique in [44] is used for the data that require a data protection technique without loss of information content. MONcrypt is applied only on numerical data. MONcrypt has algorithm for obfuscation and de-obfuscation. Outsourced data are obfuscated before they are uploaded to the cloud storage. Obfuscation module is used to obfuscate the users' data. Once the data are obfuscated, they are forwarded to cloud storage. During the obfuscation module, for original message a count value is generated. This value is maintained in the users' side as metadata. To retrieve the data from the cloud storage, de-obfuscation module is used to convert the obfuscated text into original plaintext at users' side. De-obfuscation module uses the metadata information to get the original text.

MONcrypt receives plaintext in a text file, for example plain.txt and produces obfuscated text in another text file, for example obfus.txt. MONcrypt does not only improves the confidentiality of the data but also it compresses the total size of the plaintext.

Discussion: We have analyzed the solution according to the experiments made in [44] and the results show that:

- Compared to the existing obfuscation techniques, MONcrypt has taken low time duration for obfuscating different size of plaintext;
- Compared to the existing de-obfuscation techniques, MONcrypt have taken low time duration for de-obfuscation different size of plaintext;
- Compared to the existing obfuscation techniques, the proposed technique has maximum percentage of security;

- MONcrypt technique gives maximum security and better performance than existing techniques while storing the data in the cloud;
- Hence, the confidentially of the data stored in the cloud is achieved.

2.3.6.4. Matrix-based Solution

Ensuring client data confidentiality in cloud computing can be achieved using data obfuscating techniques instead of encryption. In a matrix multiplication application, clients can protect their data from dishonest or curious cloud servers which perform multiplication operations on matrices without 'knowing or seeing' actual values of input matrices. In [45], the authors introduce random noise to the data, and generate several matrices randomly from each matrix in order to cloak data from cloud servers. The main idea is to mask the data as well as confuse the cloud server so it is unable to derive or guess the actual values of matrices as well as computer results.

Discussion: Many protocols have been proposed to address the data confidentiality issue related to secure outsourcing using various data obfuscation and data masking techniques. This solution is relatively simple, and the client needs much less effort to cleanse the correct result. The approach ensures data confidentiality without using any encryption. The main objectives of the approach are to confuse the cloud server as well as to make it difficult for the server or any unauthorized entity to derive or guess the actual values of matrices. It is assumed that the cloud server is unable to infer, derive or learn anything about the actual values. Since the client does not use any encryption, the approach does not require her to share any secret or key with the cloud server. All secrets are solely held by the client. The protocol requires the client computing effort maximum $O(n^2)$. The cloud server needs no additional protocol to compute the matrices apart from multiplication operations. That means, the server does not know anything about the protocol that the client uses. The results compared to other three protocols also suggest that the proposed protocol is much better than others in terms of performance of the server(s). The approach does not address the issue of data integrity. It does not provide any protection if the server cheats or modifies the matrices for malicious intentions. It is also unsure at this stage if the approach can be applicable to other non-numeric data type such as text. Our further research includes investigating how the integrity of data can be ensured, and how text data can be used in this approach.

2.4. General Assessment of the Studied Solutions

A classification alone cannot guarantee a good study of these solutions. In fact, in this section we will identify the main advantages and disadvantages of all studied solutions according to the group in which they are classified (Table 2.2). In addition, we will present in Table 2.3 their relevance in their adequate deployment model (Private cloud, Public cloud, Hybrid cloud, Community cloud). Finally, we will submit in Table 2.4 their resistance to the attacks seen above but also in Table 2.5 classify them according to whether they are suitable for centralized or decentralized cloud architectures and whether or not they use blockchain technology.

Table 2.2. Advantages and disadvantages of solutions ensuring data confidentiality in cloud computing.

Classification	Advantages	Disadvantages
Confidentiality Through Biometric Encryption	Increases the level of security during the authentication process [8, 11]. Faking Biometric information is much more difficult than information as passwords etc. Biometric input is unique for each individual [6-11].	Biometric encryption targets the confidentiality of biometric Information only [6, 10]. Affected by environmental and human circumstances [6]. Unlike passwords that can be changed, fingerprints and other biometric data are unique and once hacked or stolen, are at a high risk to be used again.
Confidentiality Through Homomorphic Encryption	**A** (cloud provider) can perform computations on **B**'s (cloud consumer) private data without learning B's secrets [12, 14-16]. Data can be decrypted less often, which is good for data confidentiality [12-14].	Extremely slow and computationally expensive (computations on large cipher texts are typically slower than computation on plaintext) [12, 14-16].
Confidentiality not by encryption only	Encryption alone assure security and privacy to user's sensitive information. It also offers anonymity and reduces any chances of fraud. So, adding another method to encryption reinforces data confidentiality [19-22, 25-28, 30-34].	Heavy computation. [27, 21, 22, 32, 34] Hard to recover data due to overprotective data access mechanisms [19, 20, 32-34].
Confidentiality according to the type of storage	All private data are stored in private cloud which means no access from the outside [37, 38].	In the case of hybrid cloud, all the private data have to be stored in private cloud, which makes the adoption of cloud computing meaningless [37, 38]. Heavy computation and storage overhead in private cloud [37, 38]. Important communication overhead between private and public cloud [38, 39].
Confidentiality Without Encryption	Since all technologies today are based on encryption systems, attackers no longer have the vision to look for anything other than to counter encrypted systems. Therefore, it will take them a long time to understand the procedure used to secure the data [44, 46, 47]. Encrypted data cannot be processed in many solutions, they must be decrypted in order to be processed and used. However, to get the data decrypted is a costly operation concerning the time and the required processing power. Dealing with unencrypted data is easier [43, 44, 46, 47].	Encryption works only on the data and cannot work on the Cloud infrastructure [42]. If the security system used is not robust enough, the cloud environment can be easily exposed to any kind of attack [43, 44].

Table 2.3. Effectiveness of the studied solutions according to their deployment model.

Deployment model	Solutions according to their appropriate deployment model
Private cloud	[25, 31, 33, 43, 44, 46]
Public cloud	[6-22, 25-34, 40, 41, 43, 44, 46, 47]
Hybrid cloud	[12-22, 25-29, 37-39, 41, 43, 44, 46]
Community cloud	[6-22, 25-29]

This second table allows us to note the effectiveness of the solutions studied according to their deployment model. It is important to know how to place these different solutions in their most appropriate deployment context in order to allow us to understand in which environment they evolve and how to contribute to improve them.

Table 2.4. Resistance of the studied solutions to existing attacks in the cloud.

	Known plaintext attack	Collusion attack	Brute force attack	Man in the Middle attack
Biometric Encryption in mobile cloud computing [6]	×	✓	✓	×
Privacy preserving biometric Identification in cloud computing [7]	×	✓	✓	×
CloudBI [8]	×	✓	✓	✓
BE to enhance confidentiality in cloud computing [9]	×	×	✓	×
Privacy preserving security using biometrics [10]	×	✓	✓	×
Cloud El Gamal and Fast cloud RSA schemes [11]	×	✓	×	✓
Fast variant cloud-Paillier scheme [12]	✓	✓	✓	×
A solution based on fully homomorphic encryption [13]	×	✓	×	×
Speedy Cloud-RSA [14]	×	✓	×	✓
Solution based on Intel SGX [15]	✓	✓	×	✓
Hpi-Secure [16]	✓	✓	×	✓
K-NN classifier for data confidentiality [17]	✓	✓	×	×
Steganography based solution [18]	×	×	✓	✓
Confidentiality preserving through encryption and obfuscation [19]	×	✓	×	✓
Attribute Based Proxy Re-encryption [20]	✓	×	✓	×
TR-MABE [21]	✓	✓	✓	✓
Privacy-Manager [24]	×	✓	✓	✓
Hierarchical classes of computing for privacy-preserving in cloud computing [25]	✓	✓	×	✓
Encryption and trust-based solution [26]	✓	×	×	✓
Blockchain for cloud-based data privacy [27]	✓	✓	✓	×
L-EncDB [28]	×	✓	×	×
CS-IBE [29]	✓	✓	✓	✓
GRBAC [30]	×	✓	×	✓

Table 2.4. Continued.

	Known plaintext attack	Collusion attack	Brute force attack	Man in the Middle attack
ISRSAC [31]	✓	×	✓	✓
Solution using Modular Encryption Standard [32]	✓	✓	×	✓
CPDA [33]	✓	✓	✓	✓
Secret sharing scheme [36]	×	×	✓	✓
Scheme for image data privacy [37]	×	×	✓	✓
DCSSP [38]	✓	✓	✓	×
CAFE [39]	✓	×	✓	✓
ARIANNA [40]	✓	×	✓	✓
NetDB2-MS [42]	✓	×	✓	×
Multiple storage solution [43]	✓	×	✓	✓
MONcrypt [44]	×	✓	×	✓
Matrix-based solution [45]	✓	✓	✓	✓

This table shows the resistance of the solutions studied to the few best-known attacks in cloud computing.

In today's internet world, it is easier to attack than to defend, so it is important to understand what are the key dangers looming over each solution in order to better arm yourself for defense. So that is why it was necessary to constitute this table to be able to place the solutions according to a certain scale of resistance to known attacks in the cloud.

Table 2.5. Classification of solutions according to whether they are suitable for centralized or decentralized cloud architectures and whether or not they use blockchain technology.

Solutions	Supported Architecture	Use of blockchain technology
Biometric Encryption in mobile cloud computing [6]	Centralized	No
Privacy preserving biometric Identification in cloud computing [7]	Centralized / Decentralized	No
CloudBI [8]	Centralized / Decentralized	No
BE to enhance confidentiality in cloud computing [9]	Centralized / Decentralized	No
Privacy preserving security using biometrics [10]	Centralized / Decentralized	No
Cloud El Gamal and Fast cloud RSA schemes [11]	Centralized / Decentralized	No
Fast variant cloud-Paillier scheme [12]	Centralized / Decentralized	No
A solution based on fully homomorphic encryption [13]	Centralized / Decentralized	No
Speedy Cloud-RSA [14]	Centralized / Decentralized	No
Solution based on Intel SGX [15]	Centralized	No
Hpi-Secure [16]	Centralized	No

Table 2.5. Continued.

Solutions	Supported Architecture	Use of blockchain technology
K-NN classifier for data confidentiality [17]	Centralized / Decentralized	No
Steganography based solution [18]	Centralized / Decentralized	No
Confidentiality preserving through encryption and obfuscation [19]	Centralized / Decentralized	No
Attribute Based Proxy Re-encryption [20]	Centralized	No
TR-MABE [21]	Centralized	No
Privacy-Manager [24]	Centralized	No
Hierarchical classes of computing for privacy-preserving in cloud computing [25]	Centralized	No
Encryption and trust-based solution [26]	Centralized	No
Blockchain for cloud-based data privacy [27]	Centralized / Decentralized	Yes
L-EncDB [28]	Centralized / Decentralized	No
CS-IBE [29]	Centralized / Decentralized	No
GRBAC [30]	Centralized / Decentralized	No
ISRSAC [31]	Centralized / Decentralized	No
Solution using Modular Encryption Standard [32]	Centralized	No
CPDA [33]	Centralized	No
Secret sharing scheme [36]	Centralized	No
Scheme for image data privacy [37]	Centralized	No
DCSSP [38]	Centralized	No
CAFE [39]	Centralized	No
ARIANNA [40]	Centralized	No
NetDB2-MS [42]	Centralized / Decentralized	No
Multiple storage solution [43]	Centralized / Decentralized	No
MONcrypt [44]	Centralized / Decentralized	No
Matrix-based solution [45]	Centralized / Decentralized	No

2.5. Conclusion

Concerns around questions of security and confidentiality of data remain the main obstacle to the adoption of the Cloud by businesses. In order to better understand security problems and especially the threats to data privacy in cloud environments, we have studied in this document several solutions concerning these problems. These few approaches seem to be relevant and provide an acceptable level of security, but are nevertheless sometimes insufficient.

Today we are experiencing a threat of a coronavirus pandemic (Covid-19) on a planetary scale. For safety measures, several countries have already taken decisions to temporarily all activities for less contact and meeting between people. So, with the confinement imposed by this virus, cloud computing is going through its biggest "stress test". Companies and students are therefore called upon today to use telework to continue their activities. However, Microsoft, whose cloud computing infrastructure is the most used on the planet, especially by businesses, sees its Teams service not resistant to the influx of simultaneous connections from Europe. This failure could not get worse. Because if one of the cloud computing giants is not immune to such a concern in the first hours of massive teleworking, it will be difficult to bet on the stability of companies whose IT resources are not as sophisticated.

Given this enormous instability that cloud technology encounters with these global connections occurred suddenly, threats to data confidentiality are becoming more and more current. Indeed, today any attacker could take the advantage of attacking the cloud for harmful purposes. Our role would therefore be, following the study made on the solutions ensuring confidentiality, to propose a scheme which would take into account the positive aspects of the studied solutions while taking care to rule out all the identified flaws. This scheme could bring a new touch to cloud technology and ensure a high level of security for data in the cloud.

References

[1]. P. Mell, T. Grance, The NIST Definition of Cloud Computing, Special Publication 800-145, *National Institute of Standards and Technology*, 2011.
[2]. Executive Cloud Survey, What IT Leaders are Worried About, *CITO Research*, 2017.
[3]. Cloud Security Report, https://www.isc2.org/resource-center/reports/2020-cloud-security-report#
[4]. B. Sumitra, C. R. Pethuru, M. Misbahuddin, A survey of cloud authentication attacks and solution approaches, *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 2, 2014, pp. 6245-6253.
[5]. D. Chen, H. Zhao, Data security and privacy protection issues in cloud computing, in *Proceedings of the International Conference on Computer Science and Electronics Engineering (ICCSEE'12)*, 2012, pp. 647-651.
[6]. K. Zhao, H. Jin, D. Zou, G. Chen, W. Dai, Feasibility of deploying biometric encryption in mobile cloud computing, in *Proceedings of the 8th China Grid Annual Conference*, 2013, pp. 28-33.
[7]. J. Yuan, S. Yu, Efficient privacy-preserving biometric identification in cloud computing, in *Proceedings of the IEEE INFOCOM*, 2013, pp. 2652-2660.
[8]. Q. Wang, S. Hu, K. Ren, M. He, M. Du, Z. Wang, CloudBI: Practical privacy-preserving outsourcing of biometric identification in the cloud, *Lecture Notes in Computer Science*, Vol. 9327, pp. 186-205.
[9]. M. N. Omar, M. Salleh, M. Bakhtiari, Biometric encryption to enhance confidentiality in cloud computing, in *Proceedings of the International Symposium on Biometrics and Security Technologies (ISBAST'14)*, 2014, pp. 45-50.
[10]. S. Kumar, S. Kumar Singh, A. Kumar Singh, S. Tiwari, R. S. Singh, Privacy preserving security using biometrics in cloud computing, *Multimedia Tools and Applications*, Vol. 77, Issue 9, pp. 11017-11039.

[11]. K. El Makkaoui, A. Beni-Hssane, A. Ezzati, Cloud-ElGamal and fast Cloud-RSA homomorphic schemes for protecting data confidentiality in cloud computing, *International Journal of Digital Crime and Forensics*, Vol. 11, Issue 3, pp. 90-102.

[12]. K. E. Makkaoui, A. Ezzati, A. Beni-Hssane, S. Ouhmad, A swift Cloud-Paillier scheme to protect sensitive data confidentiality in cloud computing, *Procedia Computer Science*, Vol. 134, pp. 83-90.

[13]. F. Zhao, C. Li, C. F. Liu, A cloud computing security solution based on fully homomorphic encryption, in *Proceedings of the 16th International Conference on Advanced Communication Technology (ICACT'14)*, 2014, pp. 485-488.

[14]. K. El Makkaoui, A. Beni-Hssane, A. Ezzati, Speedy Cloud-RSA homomorphic scheme for preserving data confidentiality in cloud computing, *Journal of Ambient Intelligence and Humanized Computing*, Vol. 10, 2019, pp. 4629-4640,

[15]. D. P. Zegzhda, E. S. Usov, A. V. Nikol'skii, E. Y. Pavlenko, Use of Intel SGX to ensure the confidentiality of data of cloud users, *Automatic Control and Computer Sciences*, Vol. 51, Issue 8, 2017, pp. 848-854.

[16]. E. Saleh, C. Meinel, HPISecure: Towards data confidentiality in cloud applications, in *Proceedings of the 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid'13)*, 2013, pp. 605-609.

[17]. M. Ali Zardari, L. T. Jung, N. Zakaria, K-NN classifier for data confidentiality in cloud computing, in *Proceedings of the International Conference on Computer and Information Sciences (ICCOINS'14)*, 2014, pp. 1-6.

[18]. G. S. Mahmood, D. J. Huang, B. A. Jaleel, A secure cloud computing system by using encryption and access control model, *Journal of Information Processing Systems*, Vol. 15, Issue 3, 2019, pp. 538-549.

[19]. L. Arockiam, S. Monikandan, Efficient cloud storage confidentiality to ensure data security, in *Proceedings of the International Conference on Computer Communication and Informatics (ICCCI'14)*, Jan. 3-5, 2014, Coimbatore, India, pp. 1-5.

[20]. G. Wang, Q. Liu, J. Wu, Hierarchical attribute-based encryption for fine-grained access control in cloud storage services, *in Proceedings of the 17th ACM conference on Computer and communications security (CCS'10)*, 2010, pp. 735-737.

[21]. J. Zhou, Z. Cao, X. Dong, X. Lin, TR-MABE: White-box traceable and revocable multi-authority attribute-based encryption and its applications to multi-level privacy-preserving E-healthcare cloud computing systems, in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'15)*, 2015, pp. 2398-2406.

[22]. X. Boyen, B. Waters, Full-domain subgroup hiding and constant size group signatures, *LNCS*, Vol. 4450, 2007, 1C15.

[23]. X. Liang, Z. Cao, J. Shao, H. Lin, Short group signature without random oracles, *LNCS*, Vol. 4861, 2007, 69C82.

[24]. M. Mowbray, S. Pearson, Y. Shen, Enhancing privacy in cloud computing via policy-based obfuscation, *The Journal of Supercomputing*, Vol. 61, Issue 2, 2010, pp. 267-291.

[25]. M. Van Dijk, A. Juels, On the impossibility of cryptography alone for privacy-preserving cloud computing, in *Proceedings of the 5th USENIX Conference on Hot Topics in Security (HotSec'10)*, August 2010, pp. 1-8.

[26]. Dr. S. S. Manikandasaran, Cloud computing with data confidentiality issues, *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 5, Issue 1, January 2016, pp. 97-100.

[27]. C. Esposito, A. De Santis, G. Tortora, H. Chang, K.-K. R. Choo, Blockchain: A panacea for healthcare cloud-based data security and privacy?, *IEEE Cloud Computing*, Vol. 5, Issue 1, 2018, pp. 31-37.

[28]. J. Li, Z. Liu, X. Chen, et al., L-EncDB: A lightweight framework for privacy-preserving data queries in cloud computing, *Knowledge-Based Systems*, Vol. 79, 2015, pp. 18-26.

[29]. A. Bentajer, M. Hedabou, K. Abouelmehdi, S. Elfezazi, CS-IBE: A data confidentiality system in public cloud storage system, *Procedia Computer Science*, Vol. 141, 2018, pp. 559-564.
[30]. M. Alam, N. Emmanuel, T. Khan, Y. Xiang, H. Hassan, Garbled role-based access control in the cloud, *Journal of Ambient Intelligence and Humanized Computing*, Vol. 9, Issue 4, 2017, pp. 1153-1166.
[31]. M. Thangavel, P. Varalakshmi, Improved secure RSA cryptosystem for data confidentiality in cloud, *International Journal of Information Systems and Change Management*, Vol. 9, Issue 4, 2017, pp. 261-277.
[32]. A. Shabbir, M. Shabbir, M. Rizwan, F. Ahmad, Ensuring the confidentiality of nuclear information at cloud using modular encryption standard, *Security and Communication Networks*, Vol. 2019, 2019, 2509898.
[33]. J. Wu, Y. Li, T. Wang, Y. Ding, CPDA: A confidentiality-preserving deduplication cloud storage with public cloud auditing, *IEEE Access*, Vol. 7, 2019, pp. 160482-160497.
[34]. S. Halevi, D. Harnik, B. Pinkas, A. Shulman-Peleg, Proofs of ownership in remote storage systems, in *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS'11)*, Chicago, IL, USA, Oct. 2011, pp. 491-500.
[35]. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song, Provable data possession at untrusted stores, in *Proceedings of the ACM Conference on Computer and Communications Security (CCS'07)*, Alexandria, VA, USA, 2007, pp. 598-609.
[36]. A. Butoi, N. Tomai, Secret sharing scheme for data confidentiality preserving in a public-private hybrid cloud storage approach, in *Proceedings of the IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC'14)*, pp. 992-997.
[37]. S. Yu, C. Wang, K. Ren, W. Lou, Achieving secure, scalable, and fine-grained data access control in cloud computing, in *Proceedings of the IEEE INFOCOM*, 2010, pp. 1-9.
[38]. A. Butoi, M. Moca, N. Tomai, Data confidentiality in cloud storage protocol based on secret sharing scheme: A brute force attack evaluation, in *Proceedings of the 9th IFIP International Conference on Trust Management (TM)*, May 2015, Hamburg, Germany, pp. 177-184.
[39]. S. Park, C. H. Kim, J. Rhee, J.-J. Won, T. Han, D. Xu, CAFE: A virtualization-based approach to protecting sensitive cloud application logic confidentiality, *IEEE Transactions on Dependable and Secure Computing*, Vol. 17, Issue 4, 2020, pp. 883-897.
[40]. R. Di Pietro, M. Scarpa, M. Giacobbe, A. Puliafito, An approach to enhancing confidentiality and integrity on mobile multi-cloud systems: The "ARIANNA" experience, in *Proceedings of the Fifth International Conference on Internet of Things: Systems, Management and Security (IoTSMS'18)*, 2018, pp. 244-249.
[41]. R. Di Pietro, M. Scarpa, M. Giacobbe, A. Puliafito, Secure storage as a service in multi-cloud environment, in Ad-hoc, Mobile, and Wireless Networks (A. Puliafito, D. Bruneo, S. Distefano, F. Longo, Eds.), *Springer International Publishing*, 2017, pp. 328-341.
[42]. A. Lzain, E. Pardede, Using multi shares for ensuring privacy in Database-as-a-Service, in *Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS'11)*, 2011, pp. 1-9.
[43]. Y. Shi, Data security and privacy protection in public cloud, in *Proceedings of the IEEE International Conference on Big Data (Big Data'18)*, 2018, pp. 4812-4819.
[44]. S. S. Manikandasaran, Dr. L. Arockiam Lawrence, S. K. Malarchelvi, MONcrypt: a technique to ensure the confidentiality of outsourced data in cloud storage, *International Journal of Information and Computer Security*, Vol. 11, Issue 1, 2019, pp. 1-16.
[45]. K. Khan, M. Shaheen, Y. Wang, Data confidentiality in cloud-based pervasive system, in *Proceedings of the 2nd International Conference on Internet of things, Data and Cloud Computing (ICC'17),* 2017, 185.

Advances in Networks, Security and Communications, Volume 3

Sergey Y. Yurish, Editor

Thy 3rd volume of *'Advances in Networks, Security and Communications'* open access Book Series contains 5 chapters written by 7 authors from 3 countries: Austria, India and Senegal. Each of the chapters contains the extended state-of-the art followed by new, obtained by the authors results, unpublished before.

Like the first two volumes of this Book Series, the 3rd volume also has been organized by topics of high interest. Every chapter in the volume is independent and self-contained and has the same structure: first an introduction to specific topic under study; second particular field description including applications, ending by well selected list of references with books, journals, conference proceedings and web sites.

With this unique combination of information in each volume, the in *'Advances in Networks, Security and Communications'* Book Series will be useful for post-graduate students, researchers, engineers and scientist working in the appropriate areas.

ISBN 978-84-09-47239-0



