cryptography

MDPI

Article

A Searchable Encryption Scheme with Biometric Authentication and Authorization for Cloud Environments

Marius Iulian Mihailescu [1,*] and Stefania Loredana Nita [2]

[1] Scientific Research Center in Mathematics and Computer Science, Spiru Haret University of Bucharest, 030045 Bucharest, Romania
[2] Department of Computers and Cyber Security, Ferdinand I Military Technical Academy, 050141 Bucharest, Romania; stefania.nita@mta.ro
* Correspondence: m.mihailescu.mi@spiruharet.ro

Abstract: Cloud computing offers the possibility of providing suitable access within a network for a set of resources. Many users use different services for outsourcing their data within the cloud, saving and mitigating the local storage and other resources involved. One of the biggest concerns is represented by storing sensitive data on remote servers, which can be found to be extremely challenging within different situations related to privacy. Searchable Encryption (SE) represents a particular case of Fully Homomorphic Encryption (FHE) and at the same time represents a method composed from a set of algorithms meant to offer protection for users' sensitive data, while it preserves the searching functionality on the server-side. There are two main types of SE: *Searchable Symmetric Encryption (SSE)*, where the ciphertexts and trapdoors for searching are performed using private key holders, and *Public Key Searchable Encryption (PKSE)*, in which a specific number of users have the public key based on which are capable of outputting ciphertexts and giving the possibility of producing the trapdoors by using the private key from the holder. In this article, we propose a searchable encryption system that uses biometric authentication. Additionally, biometric data are used in the trapdoor generation process, such that an unauthorized user cannot submit search queries. The proposed system contains three components: classic user authentication (based on username, password, and a message with a code using short message service (SMS), biometric authentication, and the searchable encryption scheme. The first two components can be seen as two-factor authentication (2FA), and the second component represents the initialization step of the searchable encryption scheme. In the end, we show and demonstrate that the proposed scheme can be implemented with success for medium to complex network infrastructures. We have granted special attention to the trapdoor function, which generates a value that can be used to perform the search process and search function that is based on the trapdoor pair for searching within the index structure. We provide the correctness and security proof of the operations, which gives us the guarantee that the cloud servers return the correct documents. Additionally, we discuss measuring the performance of the authentication scheme in terms of performance indicators, introducing two indicators for measuring purposes—namely, *cloud average number of non-legitim the user actions for cloud purposes* (C_{ANNL}) and *cloud average number of legitim user actions* (C_{ANLU}).

Keywords: applied cryptography; theoretical cryptography; information security; cybersecurity; searchable encryption

Citation: Mihailescu, M.I.; Nita, S.L. A Searchable Encryption Scheme with Biometric Authentication and Authorization for Cloud Environments. *Cryptography* **2022**, *6*, 8. https://doi.org/10.3390/cryptography6010008

Academic Editors: Cheng-Chi Lee, Mehdi Gheisari, Mohammad Javad Shayegan, Milad Taleby Ahvanooey and Yang Liu

Received: 11 January 2022
Accepted: 10 February 2022
Published: 14 February 2022

1. Introduction

The last two years have brought unexpected challenges in many ways. The pandemic has forced many activities to be relocated online, and in many cases, this relocation has been carried out brutally; without prior preparation, organizations are being forced by the unusual context around the world. Thus, various private information and documents were moved to the online environment to facilitate remote work. Such a large amount

of data moved in the online environment, as well as the remote work itself, led to the intensification of cyberattacks. A report released by ENISA (European Union Agency for Cybersecurity) shows that between April 2020 and July 2021 different types of attacks have been launched, which included malware, cryptojacking, or ransomware; another report [1] shows that cyberattacks increased by 29% in 2021.

To overcome these attacks, the online software systems used in different areas should implement a strong authentication process and a method of use that leaks as little information as possible. A strong authentication process may consist of two-factor authentication or even stronger biometric authentication. On the other hand, when possible, the processing of data should be made in an encrypted format, such that neither the server that makes the processing nor other entities have access in any way to the data. Moreover, the result of processing should be transmitted to the user in an encrypted format, and to decrypt them, only the user owns a decryption key. An example of such a process is searchable encryption, an encryption category that allows search operations to be made over the encrypted data, based on some search criteria. Searchable encryption can have practical application basically in any system that queries a structure/system of data.

The main objective of the current manuscript is to introduce a new approach for the authentication process, combining a two-factor authentication mechanism and searchable encryption for user authentication in a cloud environment. Searchable encryption is used to gain access to the documents based on searching functionality using keywords. The proposed authentication scheme is a combination of three important components, such as *Classic Authentication*, *Two Factor Authentication (2FA) with Biometric Characteristics*, and *Searchable Encryption*.

The core of the system lies in two modules, which use biometric authentication and searchable encryption, as follows:

- Authentication—the first module has two main components: two-factor authentication and biometric authentication. Each component implements specific services that allow a user to authenticate into the system, and each has different functionalities. Each component is given a sequence diagram that highlights the whole process, and which explains in detail how these are used.
- Searchable encryption scheme—the second module implements a searchable encryption scheme that uses the RSA cryptosystem [2] in its components and makes use of the homomorphic properties of RSA in the trapdoor generation process and search process. The scheme makes use of the user's Biometric Identity Record (BIR) or biometric template (BT) for trapdoor generation, adding an extra level of security, which enables only authorized users to search based on a query keyword. A biometric template represents a digital reference of different characteristics that have been generated based on the biometric sample. The trapdoor is exchanged between the user and the cloud server via a digital certificate issued by a trusted authority.

The contributions can be summarized as follows:

- The system presents an improvement in the 2FA procedure by executing its functionality in such a way as to accept biometric characteristics that can be used later in the process of executing trapdoor functions specific to the searchable encryption scheme (see Figure 2 and Section 2.3 for more details).
- The system includes a 2FA approach and Searchable Encryption Scheme and shows how these two components can work together to gain access to the documents without it being necessary to decrypt them, only using keywords (see Figure 3 for more details and Section 2.3 for explanations). A comparison of several authentication schemes with the proposed idea points out the main advantages and disadvantages (see Section 4.3).
- Security analysis is provided, which shows us that the proposed scheme is secure enough for its purpose (see Section 5).

- Performance analysis points out how efficient the scheme is during different types of attacks that are classified in three scenarios: inside network attacks, outside network attacks, and combined attacks.
- The remainder of the paper is organized as follows:
- Section 1—Introduction. This section gives a short introduction in which we place the current research within the actual context in which cybersecurity plays an important role.
- Section 2—Preliminaries. This section provides an overview of the elements used in the proposed system—namely, the two-factor authentication process, biometric authentication, and searchable encryption.
- Section 3—State-of-the-Art Approach. The section presents the state-of-the-art for regarding similar approaches for searchable encryption.
- Section 4—The Proposed Scheme. The section presents in detail all components of the proposed system, discussing the methods of implementation and the entities involved within the system, while the fourth section discusses the security of the system, its performance, and other practical considerations.
- Section 5—Security Analysis. This section discusses the security analysis and proof. It describes all the required computations needed to demonstrate the security of the proposed authentication scheme.
- Section 6—Performance Analysis. The section's purpose is to point out the main advantages and disadvantages of our scheme, by comparing our scheme with other existent schemes, and showing how well the proposed scheme performs.
- Section 7—Discussion. The section concludes and summarizes the results achieved. It presents proactive comments based on the results obtained and discusses further approaches on how better results might be achieved.
- Section 8—Conclusions. The section summarizes and describes the main challenges that were experienced during the process of elaborating the current research.

2. Preliminaries

2.1. Searchable Encryption

Searchable encryption (SE) is an encryption mechanism that allows search operations over encrypted data. There are two types of SE schemes: Symmetric Searchable Encryption (SSE) and Public-Key Searchable Encryption (PKSE). The difference between them lies in the number of keys used within the system: SSE uses just one type of key—namely, the secret (private) key k_s used for both encryption and decryption—while PKSE uses two types of keys—namely, a public key k_p used to encrypt the data and the secret key k_s used to decrypt the data [3]. There are two types of objects at work in SE: document files or encrypted databases.

In a system that implements SE schemes, there are more entities involved, which are described as follows [4]:

- *Trusted authority* (TA). This is an optional entity whose purpose is to generate the parameters of the system, to generate digital certificates, etc.
- *Data owner* (DO). This entity owns the data, and usually, it encrypts the data and sends the data to the cloud server.
- *Data user* (DU). This entity queries the data by computing a trapdoor value based on the keywords for which DU desires the corresponding documents and decrypts the data received from the server. Note that a data owner may be a data user.
- *Cloud server* (CS). Depending on the system architecture, there may be more servers involved in the system. However, there exists at least one server, which stores the data, receives trapdoor values that are used in the search process, runs the search process to find the documents that match the search criteria and returns the results to the data user.

In general, SE schemes consist of the following algorithms: setup, key generation, build index, encryption, trapdoor, search, and decryption. Depending on SSE's type, the algorithms have different parameters, involving the inclusion of the keys. The system

proposed in this manuscript uses a PKSE scheme; therefore, we detail below the algorithms for this type of SE scheme.

Setup. This algorithm generates the parameters of the system (SP) based on a security parameter, usually called λ. TA usually runs this algorithm.

Key generation. This algorithm is used to generate the keys of the system—namely, the secret key k_s and the public key k_p—based on the SP generated previously. The key generation algorithm is run by the DO.

Encryption. This algorithm is used to encrypt the plain documents using the public key generated in the previous algorithm and is run by the data owner. In this phase, DO sends the encrypted documents to CS.

BuildIndex. This algorithm is used to create the index structure, which contains the keywords that describe the documents. The data owner runs this algorithm, which sends the obtained index structure on a cloud server (it can be the same CS on which the encrypted data are stored). However, this algorithm is optional, depending on the design of the SE scheme. For building the index algorithm, there are two approaches:

- Classical index structure. In this approach, each document d_i is described by several keywords $K_i = \{k_{i1}, k_{i2}, \ldots, k_{im}\}$, $m \geq 1$.
- Inverted index. In this approach, each keyword k_i describes a set of documents $D_i = \{d_{i1}, d_{i2}, \ldots, d_{in}\}$, $n \geq 1$.

Trapdoor. This algorithm uses one or more keywords to compute a value called trapdoor that is sent to the server. In general, primary proposed SE schemes supported just one query keyword at a time, but at the present moment, more SE schemes support more query keywords in the same query session. The trapdoor algorithm is run by the DU using the public key or its private key, depending on how the trapdoor is designed.

Search. This algorithm uses the trapdoor value to scan the index structure and to find the documents that match the search criteria. If there are such documents, then, these are sent to the DO that triggered the search process; otherwise, a proper message is displayed. The search algorithm is run by the CS.

Decryption. This algorithm is used to decrypt the documents returned through the search process. DU runs this algorithm using its private key.

Searchable encryption is an important category of encryption schemes, as it has applicability in many domains that work with documents or (encrypted) databases. SE is a particular case of fully homomorphic encryption, an even more powerful type of encryption.

2.2. Biometric Authentication

The identification of users by utilizing user IDs, ID cards, passwords, tokens, or other elements of user identification has always represented an important challenge in designing a complex system. This kind of technique has been proven to be easily corrupted through different attacks. We have seen different attacks take place with success—the password being easily guessed due to user carelessness with regard to password security, ID cards that can be easily stolen, and so on [5].

Biometrics represents the process of being able to identify a person's identity based on their physiological and behavioral characteristics, managing the process of determining the difference between an authorized person and a fake or unauthorized one. Within our research, we started from the idea that the system should be able to identify a person based on their data, by querying through all the characteristics and checking it against those characteristics already stored in the database as hash values (note that there is nothing stored in clear in our database). Finding a person is accomplished using two directions: identification and verification. The enrolment of the users in the current system is carried out in eight phases (see Figure 1), with respect to the idea proposed by [6]. These phases can be summarized as follows: (1) capture the biometrics data, (2) process, enroll and extract the biometric template, (3) store the data in different tokens, (4) conduct real-time scanning of the chosen biometric, (5) process the biometric data and extract the biometric template, (6) match the biometric with the one saved in the database, (7) obtain the matching score,

which is sent to the application that is found within the production, and (8) record the audit training process concerning the entire authentication system.

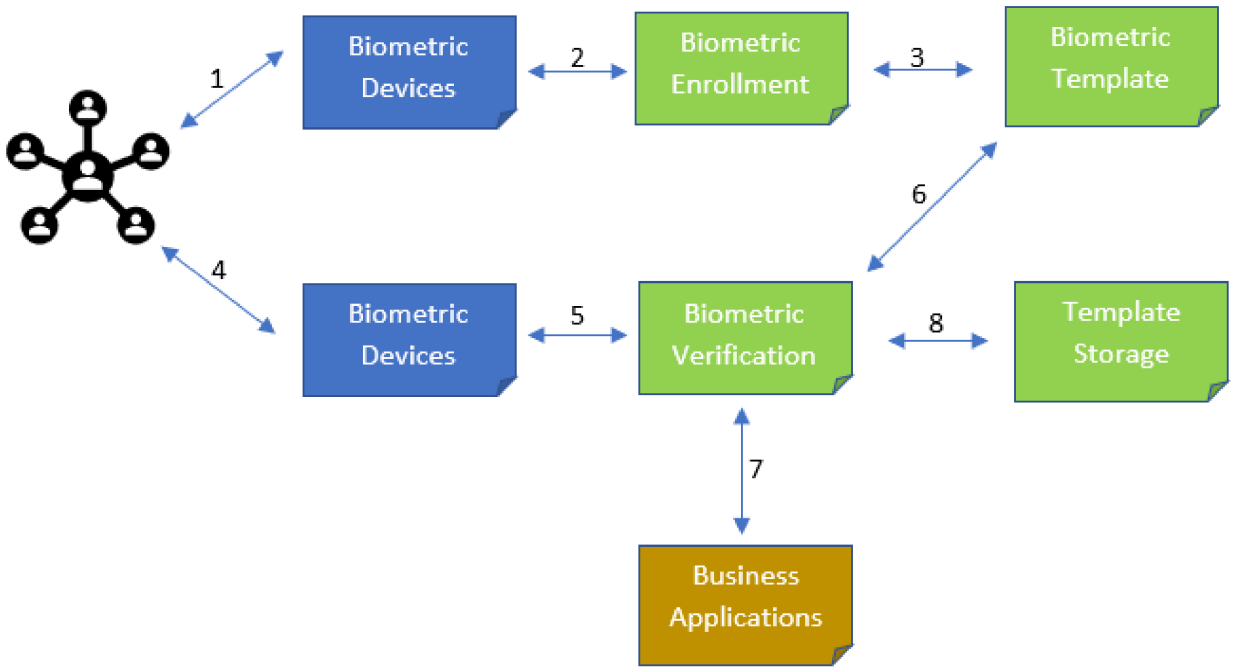

Figure 1. The Biometric Enrolment Process.

In addition to the enrolment process of known persons around us, who have served as users for testing purposes, we extended the support of biometric authentication for different types of databases, such as IEEE Biometric Databases [7] and Biometric Dataset Collections [8]. This can be very helpful for offering to our system a complex database through which the queries are performed, and different metrics are measured, such as processing time.

2.3. Two-Factor Authentication and Multi-Factor Authentication

Two-factor authentication (2FA) and multi-factor authentication (MFA) represent an authentication mechanism in which a user is asked to perform the second method of authentication once an initial login based on a username and password has been performed. In many of the applications (web software applications, mobile applications, or applicable applications for different fields [9–11]), 2FA has become critical and vital in the fight against hacking. Many accounts are hacked every day and exposed on the *Dark Web.*

The main idea of our 2FA approach is described in Figure 2. Our approach is based on *something that the user knows* (username and password), *something that the user has* (smartphone for the code received through an SMS on the smartphone), and *something that the user is* (biometric characteristic, e.g., fingerprint, facial recognition). If one looks at the BAM (Biometric Authentication Module) from Figure 2, in a quick search on the Internet one finds a wide variety of available tools. It is important to examine their features and to identify their advantages and disadvantages. Examining security flows represents another criterion that should be used when choosing such a tool (open source or not). The examination can be performed using platforms such as CVE (http://cve.mitre.org/, access on 10 January 2022) or ATT&CK (https://attack.mitre.org/, access on 10 January 2022), both tools from MITRE. Many guidelines and exploits/vulnerabilities can be identified using this platform, and for some of them, many patches have been uploaded as a solution for fixing them.

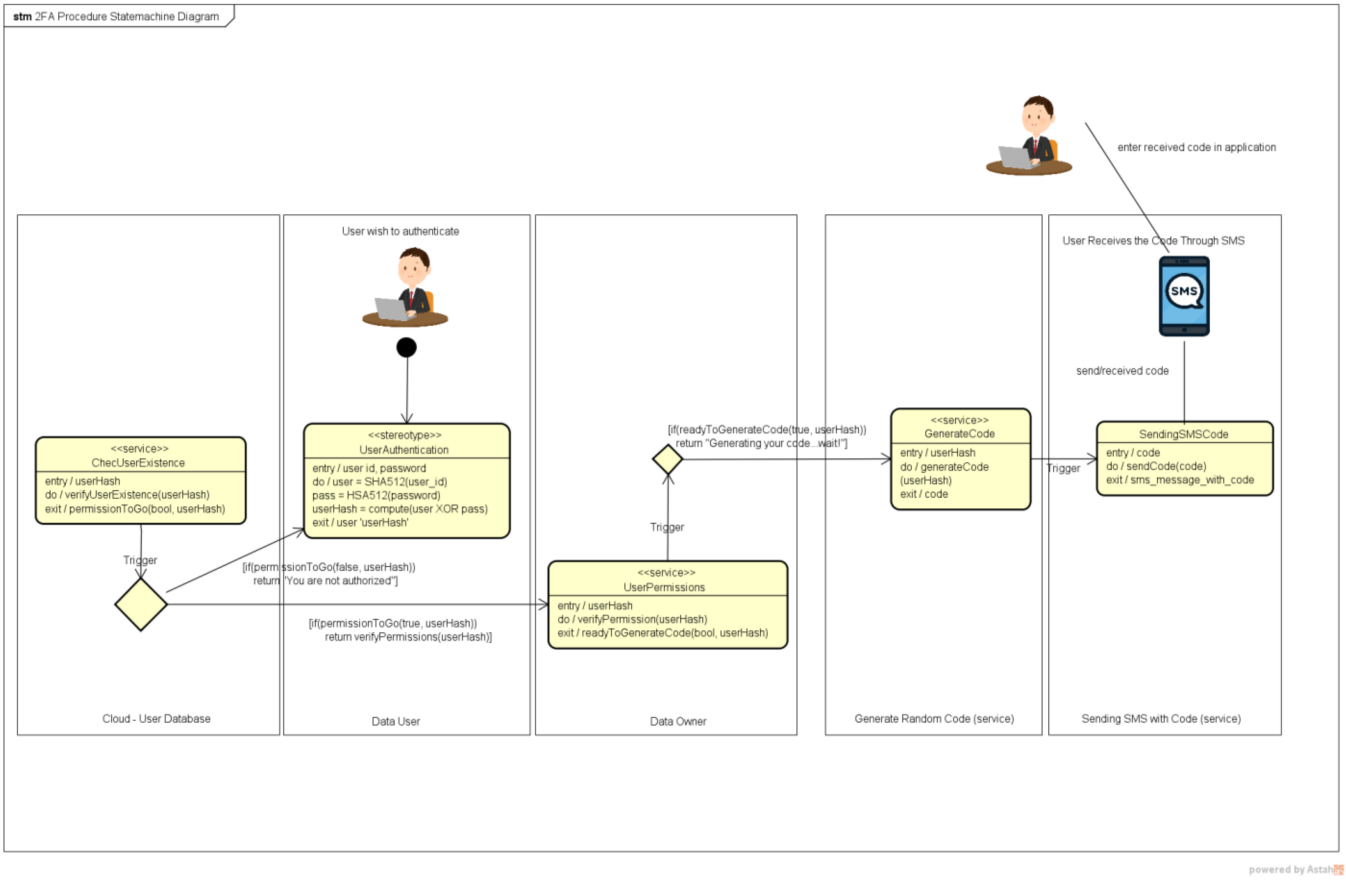

Figure 2. The state-machine diagram of the 2FA procedure and BAM.

Testing a 2FA scheme is not a trivial procedure (see Figure 2). A 2FA scheme is characterized by two important components that cannot be controlled easily or commonly (already known by different tools or procedures), such as randomness algorithm/method and the device/service/server (used to generate the codes/tokens) or an entire infrastructure of servers with different services. The design process of the 2FA scheme has taken into consideration both, the randomness algorithm, and the infrastructure of servers, fulfilling what is necessary from the security point of view, as follows:

- *Randomness*—Through the generation of a unique code received on a user's smartphone, we make the permission requirement hard to predict, it is a vital part of the testing process. We started from the idea of predicting the output in such a way as to be able to check our expectations with real output.
- *The infrastructure of servers with/or services*—If we are using external tools or services, such as the 2FA service from Google (Mountain View, CA, USA) or Microsoft (Redmond, WA, USA) or external devices, it means that we need to pay attention to security because we are dealing with something that is out of our control. Performing tests against such infrastructure or devices can have high costs or testing may not be available within your environment, especially due to the continuous process that is represented by an SMS message or communication subscriber.

In Figure 2 we depict the *BAM* model, which is composed of five modules, such as:

- *Cloud—User database.* The module is responsible for the interaction of the user with the database that holds information about the user and much more. The database is stored in cloud computing. For this scenario, we chose the Microsoft Azure platform (Redmond, WA, USA). In Microsoft Azure, we deployed the *CheckUserExistence* service, which runs continuously and checks each user who wishes to authenticate.

- *Data user.* The module is in charge of the authentication process and service for verifying the user identity.
- *Data owner.* This module verifies for each user all the permissions that are available for the user and decides terms of acceptance.
- *Generate random code (service).* Based on the user identity, the service computes a hash value of the user identity, and using the hash value, a code is generated and passed further to the *Sending SMS with the code* module.
- *Sending SMS with code (service).* This module sends a generated code to the user's smartphone to be used within the authentication process

The main functions that describe the *BAM* module related to our proposed scenario from Figure 2, are as follows:

- *CheckUserExistence* represents a service that verifies whether the user exists as enrolled or not. The service runs when the user wishes to authenticate.
- *UserAuthenticate* class represents the main point of interaction between user credentials as identity, verifying user identity with the help of the *CheckUserExistence* service, and *UserPermissions* service.
- *UserPermissions* represents a service that runs to check the permissions of a user. The permissions are verified using a hash value of the user identity, identified by the *userHash* variable.
- *GenerateCode* represents a service that is based on the user hash value obtained from the *UserPermissions* service, which generates a code, and with the help of the *SendingSMSCode* service, it is sent further to the user's smartphone.
- *SendingSMSCode* represents the final step, which is responsible for obtaining the code from the *GenerateCode* service to send it to the user and display it on the user's smartphone.

Functional testing represents a procedure used in the validation process of the software modules against the functional requirements or specifications. The goal of the functional tests is to take each function or method from a software module or software application, provide the proper input, and based on the input, verify the output against the functional specifications and/or requirements. Additionally, in the functional testing we used for our 2FA scheme, the black box testing is not focused on the source code of the application. The black box testing includes different verifications focused on security, distributed communication, APIs, and GUIs. The process used for the current state of research was performed manually. Automation procedures for testing represent a future research direction.

In Appendix A, we provide the unit tests that we wrote for our 2FA mechanism for a scenario in which the user introduces the wrong credentials. We considered that the 2FA service would be represented as an external system/service/server. The unit tests were written in C#, and we tried to keep them as customizable as possible.

In Figure 3, we present a general overview of the 2FA approach and how it can be integrated with a searchable encryption scheme. The 2FA scheme has two main modules that represent the main core of the authentication scheme, such as *Classic Authentication (UAM Module)* and *2FA Authentication (SMS code, Biometric Characteristic)—BAM Module.* Both types of authentications are involved in the process of identifying the identity of the user and initializing the searchable encryption scheme (see blue square entitled *Initializing the Searchable Encryption Scheme).* Each user's identity is verified based on a set of parameters, such as *username*, *password*, *SMS code*, and *biometric characteristic* (which can be fingerprint or facial scanning). Additionally, the user can reset his password or recover his entire account.

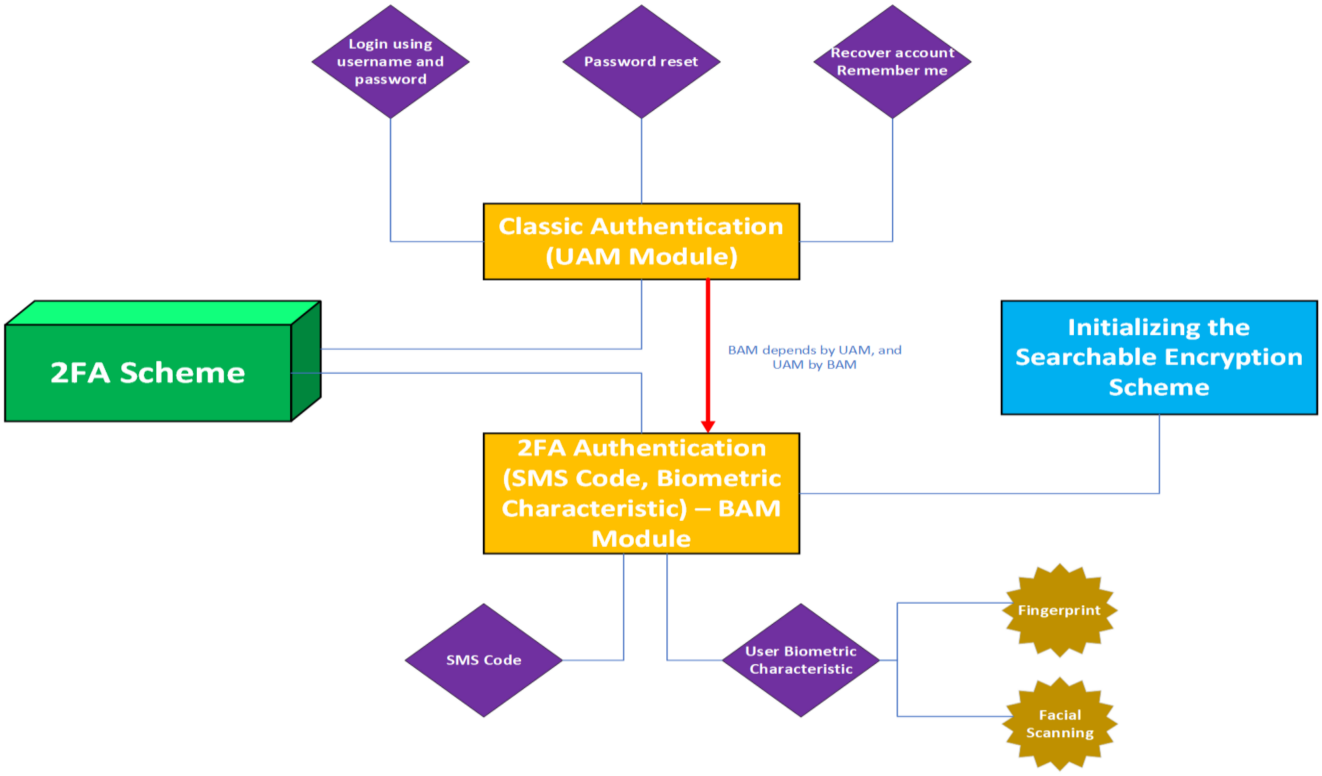

Figure 3. The main idea behind the 2FA approach and Searchable Encryption Scheme.

3. State-of-the-Art Approach

The research regarding authentication types provides very well-organized surveys in which different approaches for biometric authentication are presented, as well as presenting the challenges raised by this authentication technology [12]. Additional interesting papers that present authentication based on multiple factors are Refs. [13,14]. Cloud computing and big data opened the way for Internet-of-Things (IoT) devices. To secure the connection of systems within, often an IoT device receives a virtual identity, such that it is recognized and authenticated within the system to which it belongs. Such a type of authentication is presented in Refs. [15–17], while [18] presents a study on using blockchain authentication in variable message format (VMF) as a military standard. Another method of authentication is based on blockchain technology, and it can be applied to authenticate users in smart grid infrastructures [19,20]. Other interesting approaches for authentication are presented in [21], in which the authentication is based on information about body composition analysis; ref. [22] presents a method of authentication based on eye color and facial recognition; ref. [23] presents an authentication method based on hand gestures captured by surface electromyography.

For reflecting on the progress of our current research, we took into consideration the recent advances that have been accomplished in *attribute-based encryption with keyword search* [24–28] and *attribute-based encryption with privacy protection and accountability* [29–35].

In *attribute-based encryption with keyword search*—using key-policy attribute-based cryptography to generate different types of trapdoors that support different types of gates (e.g., AND, OR, and threshold)—represents one of the most novels approaches in designing searchable encryption with fined-grained access control.

In *attribute-based encryption with privacy protection and accountability*, most of the contributions are accomplished in the direction of ciphertext-policy attribute-based encryption (CP-ABE) mechanisms that provide the possibility of enabling fine-grained access control for encryption within data related to Internet-of-Things (IoT) data and the cloud. CP-ABE is one of the most challenging and shows great potential in providing flexible and fine-grained access control. This mechanism is much more suited for securing cloud ap-

plications, illustrating the complex authentication mechanisms, such as the one proposed in the current manuscript. Regarding searchable encryption, recent studies have a focus on dynamic schemes that allow operations of insert, update, or delete to be carried out directly on the encrypted data. Such studies in this direction are [36–38]. In addition, searchable encryption is used for biometric authentication—namely, the operations within the authentication process that are made directly over encrypted biometric data, without a need for decryption while applying the computations. Examples of such studies are presented in [39–41]. Additionally, searchable encryption can be applied in IoT or edge cloud environments [42] or blockchain [43,44].

4. The Proposed Scheme

The first draft of the current idea behind the proposed scheme was designed using a distributed infrastructure, and once the results were positive and we saw that its applicability could be extended properly to complex architectures, we started to adjust it to cloud computing and big data environments (see Figure 4). Components (module) 1 and 2 can be seen as a two-factor authentication scheme.

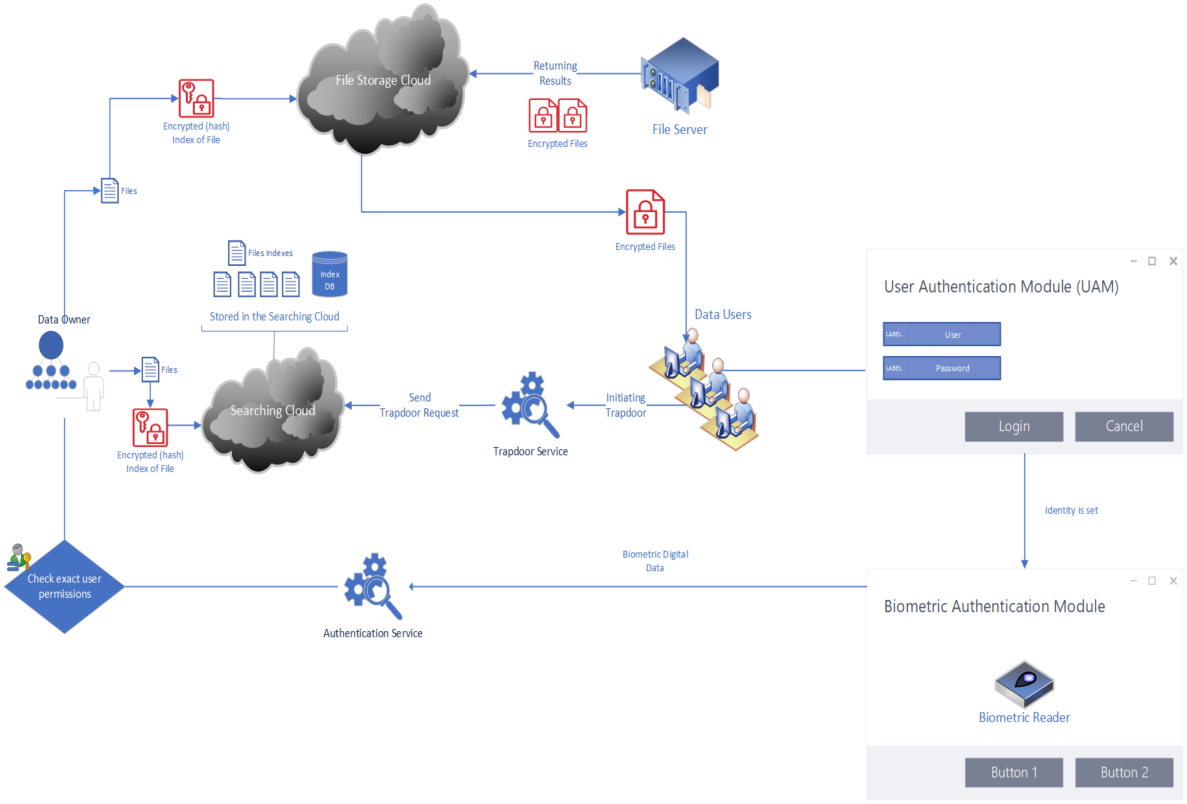

Figure 4. The entire overview of the proposed system.

The proposed scheme has three important components (or modules) that are connected and cannot function if one is not fulfilled properly. The components are:

1. *Searchable Encryption Scheme (SES)*. Based on UAM and BAM, the searchable encryption is initialized and ready to perform the searching process for the keywords entered by the users. Based on those keywords, a set of documents is returned to the user.
2. *User Authentication Module (UAM)*. Its main purpose is to identify the user that wants to authenticate with the system in a cloud environment.
3. *Biometric Authentication Module (BAM)*. This module represents an extra security measure, and it intends to certify based on one or more biometric characteristics that verify a user's proper identity.

4.1. Searchable Encryption Scheme

Searchable encryption is one of the main components of the proposed system. It uses the RSA encryption system [45] to encrypt both the documents and the keywords. Related to RSA security, the authenticity of the public key is ensured by the TA. Furthermore, the search process makes use of the homomorphic properties of the RSA cryptosystem [46] to search within the index structure. Given two messages m_1 and m_2 for the RSA cryptosystem, the homomorphic property is shown in (1), as follows:

$$Enc(m_1) \cdot Enc(m_2) - Enc(m_1 \cdot m_2) \tag{1}$$

The index structure in the proposed system lies in the category of inverted index structure—namely, there are more keywords—and each keyword has attached to it the ID of the documents that describe it. The searchable encryption scheme uses the RSA encryption system in its steps.

The entities involved in the proposed system are TA, DO, DU, and CS and each entity have the characteristics described in Section 2.1. Next, we present the algorithm of the proposed searchable encryption scheme.

Setup. The system parameters are computed based on a security parameter λ.

Key generation. In this phase, the keys for the RSA system are generated. This process is carried out each time a user is registered into the system. Therefore, each user (it can be a DO or a DU) owns a public key k_p and a secret key k_s.

Encryption. In this phase, the data owner(s) encrypt(s) the documents using the RSA encryption system and send(s) them to be stored on the CS.

BuildIndex. This algorithm is run by the DO and uses a hash table ht which is stored within the bounds between the keywords and documents. For the hash function h, it is recommended to use one represented on at least 512 bits (for example, SHA-512, or Keccak-512). Let $K = \{k_1, \ldots, k_m\}$ be the set of the keywords used within the system. Each keyword k_i has attached a set $D_i = \{d_{i1}, \ldots, d_{in}\}$, $n \geq 1$ containing the IDs of the documents on which it describes. Construction of the index structure must proceed as follows for each keyword k_i:

1. Encrypt the keyword k_i with the RSA encryption using the DO's public key. The value $c_i = Enc(k_i)$ is obtained.
2. Instantiate an empty list L.
3. For each $d_{ij} \in D_i$: encrypt d_{ij} using the RSA cryptosystem, obtaining $c_{ij} = Enc(d_{ij})$ and add c_{ij} to the list L.

Generate an entry for the hash table as follows: the key for the entry is c_i , and the value is the list L.

When the list of the keywords was consumed and the process from above was made for each keyword, the hash table is fully created and can be stored on the CS. Therefore, the DO sends the hash table to CS.

Trapdoor. The trapdoor is called by the DU, to generate a value based on which the server initiates the search process. This process is also used by the DU's BIR stored on the DU's device. The DU chooses a query keyword k_q, then it proceeds as follows:

1. Encrypt the DU's BIR using the RSA-based DO's public key and obtain $c_{BIR} = Enc(BIR)$.
2. Encrypt k_q using the RSA-based DO's public key and obtain $c_q = Enc(k_q)$.
3. Compute the value $c = c_{BIR} \cdot c_q$.
4. Compute the value BIR^{-1}, and encrypt it $t = c_{BIR^{-1}} = Enc(BIR^{-1})$.
5. Resemble the trapdoor value as the pair of $Tw = (c, t)$.
6. After the DU achieves the Tw pair, it asks the TA to release a certificate through which the trapdoor pair is sent to the CS.

Search. The server receives from the DU the trapdoor pair Tw. To search within the index structure, CS proceeds as follows:

1. Compute the value $v = c \cdot t$.

2. Search in the hash table for the value of the key v.
3. If there is such a key, then extract the value for the key; otherwise, send a proper message to the DU.
4. For the extracted key from the hash table, obtain the corresponding value, and then for each encrypted ID of a document, send that encrypted document to the DU.

Decryption. When the DU receives the encrypted documents from the server, it decrypts them using the DU's private key.

Correctness. Further, we discuss the correctness of the operations, as shown in (2), which demonstrates that the cloud server indeed returns the correct documents if these exist. The CS searches within the hash table for a key that is the encryption of a keyword. By applying the property from (1), indeed, the search is executed correctly because:

$$\begin{aligned} v = c \cdot t = c_{BIR} \cdot c_q \cdot \mathrm{c}_{\mathrm{BIR}^{-1}} &= Enc(BIR) \cdot Enc\left(k_q\right) \cdot Enc\left(BIR^{-1}\right) \\ &= Enc\left(BIR \cdot k_q\right) \cdot Enc\left(BIR^{-1}\right) \\ &= Enc\left(BIR \cdot k_q \cdot BIR^{-1}\right) \\ &= Enc\left(k_q\right) \end{aligned} \tag{2}$$

Therefore, the server obtains each time a correct key for the hash table entry.

4.2. Generation of the User ID and User Authentication Process

The authentication process (as illustrated in Figure 5) consists of the seven following steps:

1. *Step 1*—The user enters credentials (e.g., username and password).
2. *Step 1.1*—The hash value of the credentials is computed and *userHash* is returned.
3. *Step 1.1.1*—Based on the *userHash*, a token is generated and returned to the user via smartphone as an SMS, *token2*.
4. *Step 2*—The user enters the token or acknowledges it.
5. *Step 2.1*—The value of the token from the users is taken and the validation of the token is invoked.
6. *Step 2.1.1*—On the server-side, the token is validated accordingly, and a Boolean value is returned. The Boolean value returned is based on *token1* and *token2* comparison processes, as can be seen from the constraint. Both values—*token1* and *token2*—are computed as hash values using the SHA algorithm with the 512-bits size length.
7. *Step 2.1.2*—The access is granted to the user for the value that the Boolean variable has.

4.3. Biometric Authentication Process

The process of generating the user id during the user authentication process is composed of an exchange process of functions that interacts with two main components, *AuthenticationService* and *SHEBA_UserServer* (SHEBA represents the acronym of the proposed idea and system, Searchable Encryption Biometric Authentication). Let us take a closer look at the functions and how they exchange data between those two components. The process is as follows:

- Function *login(user_id,password).* This is the first function (Step 1) that is executed once the user provides the *user_id* and *password.* The function is invoked from the *AuthenticationService* that has an *EventListener* attached to the user behavior that triggers the execution of the *login* function.
- The *AuthenticationService* executes the *computeHash* (Step 1.1) function for the *user_id* and *password*, and the hash value is stored and saved as a *userHash* variable.
- On *SHEBA_UserServer,* the function *generateTokenID* (Step 1.1.1) is called for the hash value obtained in Step 1.1. The token value obtained has to be checked properly to see whether the hash value was corrupted. Observe that a string variable is defined, entitled *token2,* which holds the SHA-512 value for the *tokenID* generated before in Step 1.1.1 and returned to the *AuthenticationService* and returned further to the user's smartphone.

- In Step 2, the user enters the token value on his smartphone. The function *enterTokenID* from *AuthenticationService* receives the token value, and it sends it further to the *SHEBA_UserServer* by using the *sendTokenIDForValidation* function. In *SHEBA_UserServer,* we validate the token by comparing the two tokens: one that is obtained from the user (see Step 2.1) and the one that has been generated on the server (see Step 1.1.1). If the function *validatingTokenIDs* (Step 2.1.1) returns a bool value, and if the bool value is *true,* it means that the token provided by the user is the right one; otherwise, it will be returned as *false.*
- Based on the *true* value, in Step 2.1.2 with the help of the *grantAccessToBiometricPanel* function, we give the user access to move forward with the second phase of the authentication process, which consists in providing his biometric characteristic (fingerprint or facial).

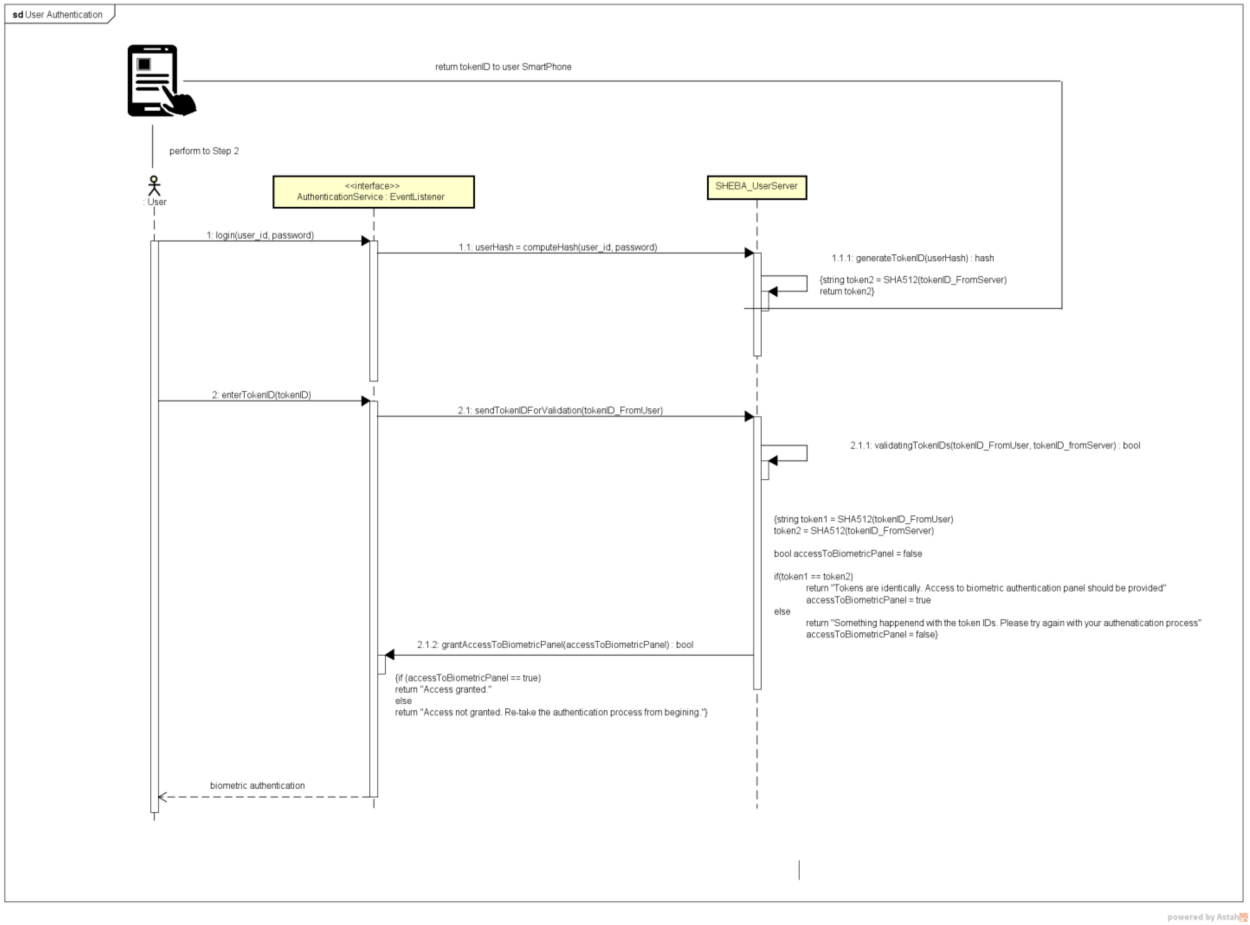

Figure 5. User Authentication Sequence Diagram.

The main objective of this sub-section is to present the user biometric authentication (UBA) component of the proposed scheme. To achieve the access of the user in the application and to initialize the searchable encryption scheme, we need to be able to verify the identity of the user, set a flag (which shows whether the user can have access), and based on the flag, initialize the searchable encryption scheme.

One of the simplest biometric authentication schemes can be described as the interaction between a user and a server. Such authentication procedures are exposed to different vulnerabilities, and it is important to avoid data in clear during the authentication process.

Based on the user authentication scheme proposed in Sub-Section 4.1, the BAM is used based on the identity of the user. The main functionalities (referring as 'F') of the proposed BAM, are described as follows (according to the sequence diagram from Figure 6):

- F1—The user identity is claimed based on the user authentication scheme presented above, in Figure 2.
- F2—Based on the user identity and the success of the authentication process, access for the user to the UBA module is granted and is performed. This granting process is accomplished using the function *grantAccessToBiometricPanel,* which receives as an argument the *accessToBiometricPanel,* a Boolean value. Depending on the value, *true* or *false*, access is granted if the value is *true*.
- F3—The user can select the biometric characteristic (e.g., fingerprint, facial recognition) by invoking the function *chooseBiometricCharacteristicType.*
- F4—The *CapturingBiometricData* service converts the biometric user data into digital data, together with the identity of the user claimed on F1. The final step of this service is to encrypt the claimed identity of the user and XOR-it with the user biometric data.
- F5—Using *UserBiometricDataDBService* we query the biometric user database (BUDB) using the user digital data and save the result in *queryResult.* The result is represented by the biometric template of the user.
- F6—In the *ComputationService* component, we invoke the *computeDistance* function for *queryResult* and *biometric_user_template* arguments. The purpose of the function is to compute the distance as being $\delta = distance\left(\widetilde{\phi}, \widetilde{\theta}\right)$, where ϕ is the user's digital data, and θ is the biometric user template.
- F7—The purpose of the *AuthenticationService* is to check the computed distance using an accuracy level using the function *checkDistance* for *distance* as an argument. As long as the δ is less than the accuracy level, the answer of the authentication server will be true.

An important aspect related to the Biometric User Database is the fact that it is stored as a database in the cloud (remember that we have mentioned that for this research we used Microsoft Azure). Two important components interact with the database—namely, *ComputationService* and *BiometricUserDatabase* itself (check Figure 6, the middle part of the figure).

The *CapturingBiometricData* module is responsible for how the biometric data are captured, converted, and stored (see Step 4.1). This is a critical step due to the complexity and role that it plays in assuring the security of the data. The process of encryption can be summarized as shown in Figure 6 under Step 4.1.

The *ComputationService* represents the component that based on the user digital data received from Step 4.1 performs the query of the database, as it is shown in Step 4.1.1. The result of the *queryUDB* function is stored in the *queryResult* variable. We compute the distance between the query result and the biometric user template that is stored in the database. The purpose is justified by functionality 7 (F7), as described above.

4.4. Comparison of Several Authentication Schemes with the Proposed Idea

Currently, most of the authentication schemes are independent of being combined with other complex security schemes, such as searchable encryption, identity-based encryption, or fully homomorphic encryption. Different approaches and examples were discussed in Section 3, and once they are properly analyzed, the advantages and disadvantages are quite clear. To provide a comparison between standard authentication mechanisms and our proposed scheme, the disadvantages of various techniques are taken into consideration and classified into a set of parameters, related to the scenarios discussed in Section 6. In Table 1 we can see a comparison of different authentication schemes using a set of parameters, such as *type of attack*, *authentication location*, *hardware/software requirement*, *brute-force*, *dictionary*, *random guessing*, *phishing*, *environment*, *extra security*, and *cryptography primitives*. Some of the parameters represent types of attacks, such as brute-force, dictionary, random guessing, and phishing, where *Yes* represents its success against that type of attack, and *No* represents that the attack was executed with success.

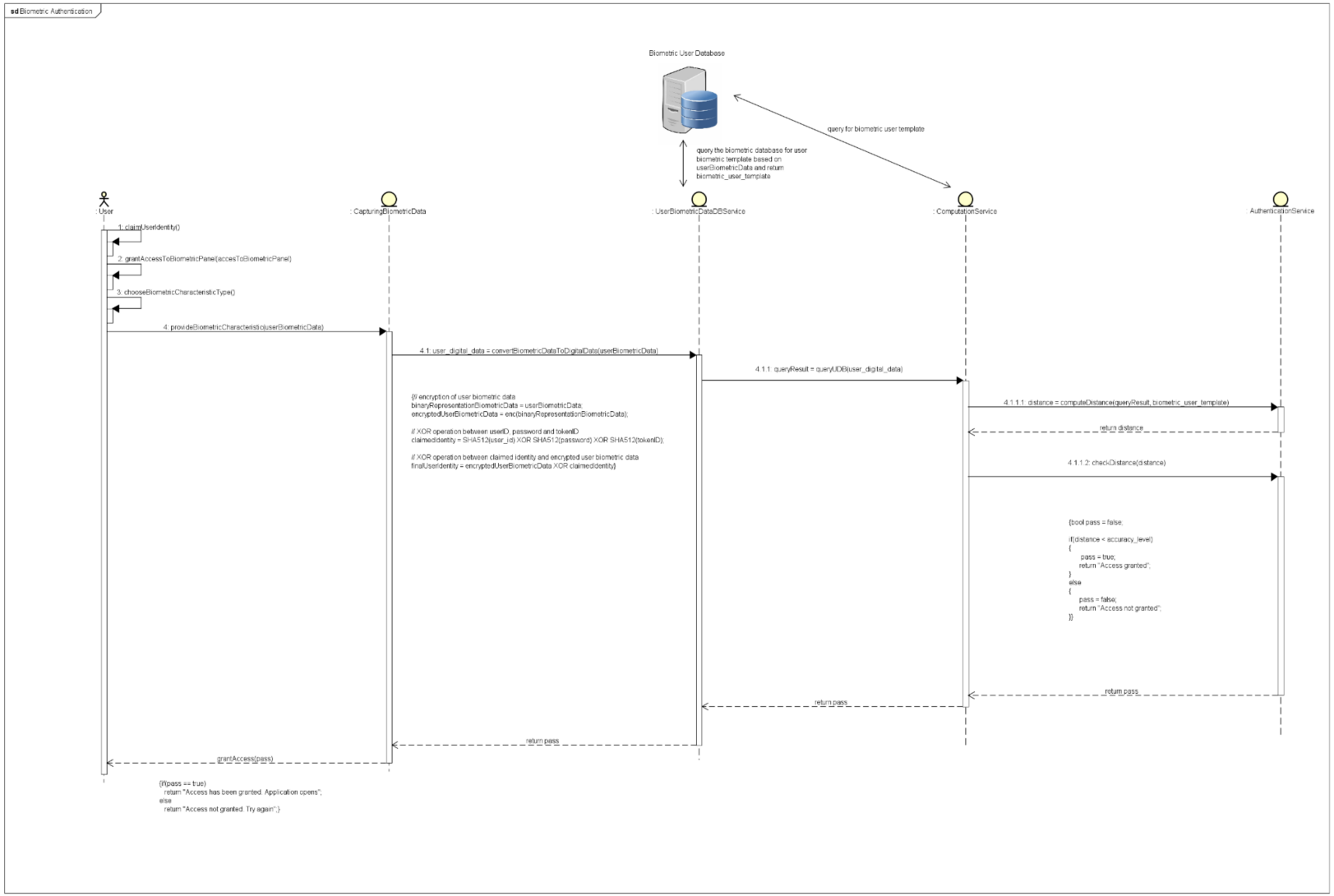

Figure 6. Biometric Authentication Sequence Diagram.

Table 1. Comparison of different authentication schemes.

Authentication Scheme	Type of Attack	Authentication Location	Hardware/Software Requirement	Brute-Force	Dictionary	Random Guessing	Phishing	Environment
The proposed scheme	Inside Outside	Server/Client	Dedicated Hardware	Yes	Yes	Yes	Yes	Cloud
Multi-factor authentication [47,48]	Inside	Client	N/A	Yes	No	No	No	Client-server
Authentication scheme using proactive model [49]	Inside Outside	Server/Client	N/A	Yes	No	Yes	Yes	Client-server
Strong user authentication [50]	Outside	Client	N/A	Yes	Yes	No	No	Cloud Client-server
Single sign-on authentication [51,52]	Outside	Client	N/A	Yes	Yes	No	No	Cloud Client-server
SOA (Service Oriented Architecture) [53,54]	Inside Outside	Server/Client	N/A	Yes	No	No	Yes	Cloud

5. Security Analysis

The security of the proposed system consists of three mechanisms: biometric authentication, incorporation of the RSA within the searchable system, and using the biometric feature of the data used for trapdoor values generation. The inclusion of biometric authentication within the system ensures that only authorized users may authenticate and use the resources. Moreover, the biometric feature itself is a part of the trapdoor generation process; therefore, to query the data, firstly, the data user needs to be authenticated, and secondly, the data user should apply his/her biometric feature to compute the trapdoor value. Further, we show that the proposed scheme is secure against an adaptive chosen keyword attack in a similar manner, as shown in [55], in which a security game is played between an attacker A and a challenger C:

- C obtains the pair of keys $(k_p,\ k_s)$, as described in Section 4.1, making publicly available k_p and keeping private k_s.
- A chooses a query keyword from the set $Kw \in \{0,1\}^*$ and asks C for the corresponding trapdoor T_w.
- A chooses two challenge keywords $kw_0,\ kw_1$ and sends them to C, with the constraint of not querying the trapdoor value for them previously. Then, C chooses randomly a value $a \in \{0,1\}$ and sends to the attacker the value $c = Encryption(k_p,\ kw_a)$.
- A may continue to ask for trapdoor values, with the constraint of not asking for Tw_0 or Tw_1.
- At some point, A makes available $a' \in \{0,1\}$, winning if $a = a'$.

The advantage of the attacker to win the game is given by the following formula from (3) [56]:

$$Adv_A(\lambda) = \left|\Pr[a = a'] - \frac{1}{2}\right|, \tag{3}$$

where λ is the security parameter of the system.

The proposed searchable encryption scheme is secure against an adaptive chosen keyword attack if the encryption is semantically secure. As the encryption is the encryption algorithm of the RSA and taking into consideration the fact that in practice RSA is implemented in the padding version (making it semantically secure), then the encryption used in our proposed scheme is semantically secure. Therefore, the proposed scheme is secure against an adaptive chosen keyword attack.

Further, we show that the proposed searchable encryption scheme is secure against the keyword guessing attack (KGA) [57]. A KGA is a type of attack that targets keywords and trapdoors. When the keywords space is relatively small (meaning a polynomial-size), an attacker can encrypt all keywords using the encryption algorithm. Then, the attacker can launch the KGA when it obtains a trapdoor value by searching within all encrypted keywords. The proposed searchable encryption scheme is secure against KGA because although an attacker gets the encrypted keywords (note that these are encrypted using the RSA algorithm), the attacker would not be able to decide whether an obtained trapdoor value corresponds to an encrypted key. This is because keywords are encrypted using the RSA algorithm (and then hashed with a specific hash function), while the structure of the trapdoor value is different, involving the DO's biometric feature. However, if the attacker manages to reach the encrypted keyword under the trapdoor, padding RSA ensures

semantic security; thus, the attacker cannot distinguish between the keyword under the trapdoor value and the encrypted keywords within the keywords space. Therefore, the proposed scheme is secure against keyword guessing attacks.

6. Performance Analysis

For performance analysis, we used three types of scenarios, from *inside network scenario*, *outside network scenario*, and *combined scenario*. The main idea behind these scenarios is based on machine learning methods and tools, in such a way that for each of the users involved in the authentication process we use and train a binary classifier. This means that the entire set of data that characterizes a user or a set of users is used for training purposes within binary classifiers and that the scheme classifies the users as *legitimate users* and *non-legitim users*. To achieve this task, each classifier is trained using a mix of a *legitimate user* and *non-legitim user* data using an equal quantity, in such a way that the bias is successfully avoided. An important aspect that needs to be mentioned is that the testing of the classifiers process is not performed with data that were used during the training process.

In the first scenario, *inside the network*, we start with the idea that the system is used inside the organization. Inside the organization, the data of all users involved in the authentication process are available. The data are used to train the binary classifiers. For example, let us suppose that if we have to deal with N users who are participating in the authentication process within the organization, then the binary classifier will be trained using data from one legitimate user and $N-1$ non-legitim users.

In the second scenario, *outside network scenario*, we start from the idea that the authentication system is used by users for whom there are no data available for training the classifier. This is concluded by providing a set of data from non-legitim users, by making the system available to different categories of users, or by externalizing the authentication module as a service or exposing it intentionally with the goal that each user should be capable of training the classifier that is based on non-legitim training data together with user own training data. In this scenario, the testing process can be achieved by dividing the non-legitim users into two sets—namely, N_1 and $N-1-N_1$ users.

In the third scenario, *combined scenario*, we deal with both types of scenarios, the *inside* and *outside* scenarios. In this scenario, we assume that N_1 represents non-legitim users from the organization and are involved in the authentication process and that their data are used within the training process. The rest of the users $N-1-N_1$ are considered to be from outside of the company, and there are no data of these users to be considered as being available during the training process.

In this work, we do not consider FMR and FNMR as the most appropriate indicators that can be used to measure the performance of the authentication system. Starting from this idea, we provide an extended version of [58] for the proposed authentication system in such a way that we extend the proposed indicators for performance analysis. The proposed indicators for measuring performance by the authors in [58] are related to the *average number of impostor actions (ANIA)* and the *average number of genuine actions (ANGA)*. Comparing their contribution, and based on their achieved results, we propose similar indicators that are meant to compute the *cloud average number of non-legitim user actions for cloud purposes* (C_{ANNL}) and *cloud average number of legitim user actions* (C_{ANLU}). The relevant difference between *ANIA* and *ANGA,* and our extended version of C_{ANNL} and C_{ANLU}, is represented by the cloud computing environment.

If a non-legitim user k tests the authentication scheme by impersonating legitim user p, after n times of unsuccessful tries, $T_1, T_2, \ldots, T_n$, non-legitim user k is blocked out. Based on this logic, we use the binary classifier (B_c) to train data of any trial that the non-legitim user might perform within the cloud environment, and our C_{ANLU} is defined in Equation (4) as follows:

$$C_{ANNL_p^k} = \frac{1}{n} \cdot \sum_{l=1}^{n} T_n \cdot B_c(T_n) \tag{4}$$

From Equation (3) we can apply C_{ANNL} against a legitimate user p in Equation (5), and we obtain the following:

$$C_{ANNLp} = \frac{1}{N-1} \cdot \sum_{l=1}^{N-1} C_{ANNL_p^k} \cdot T_{N-1} \cdot B_c(T-1) \quad (5)$$

In Equation (5), we can see that the sum is applied for all the non-legitim users, which means all users except the legitimate user.

If we apply C_{ANNL} as in Equation (6) against any legitim user for our authentication system, then C_{ANNL} can be expressed as below:

$$C_{ANNL} = \frac{1}{N} \cdot \sum_{p=1}^{N} C_{ANLUp} \cdot B_C\left(C_{ANNLp}\right) \cdot B_c(N-1) \cdot B_c(N) \quad (6)$$

Further, we define C_{ANLU} using the same logic and flow against any legitimate user who is involved in the authentication process.

7. Discussion

Security of the SES module. The proposed SE scheme fulfills the criteria defined in [59], as follows:

- Controlled searching. The fact that only authorized users submits search queries is ensured by more components. Firstly, the user must be logged into the system to submit search queries. The login process is a 2FA and then consists of biometric authentication. Secondly, the DU uses the biometric feature to generate the trapdoor value.
- Encrypted query. The pair that resembles the trapdoor is encrypted using the RSA encryption system.
- Query isolation. Using the homomorphic properties of the RSA cryptosystem, the server carries out the search process on the encrypted data; therefore, it learns nothing about the data.

Mainly, the security of the proposed searchable encryption lies in the security of the RSA cryptosystem that is used to encrypt both the index structure and the plain documents.

Biometric authentication. Based on the functionalities described in the previous section, the biometric authentication procedure is proposed as a distributed architecture that can be used and implemented with success within a cloud computing environment or big data. Biometric authentication is composed of two phases: the enrollment phase and the authentication phase. At this moment, we do not focus on the enrollment phase, and we assume that all the users are already enrolled in the system. The biometric authentication phase is adjusted in such a way that the searchable encryption scheme can be initialized properly and easily adapted to a cloud or big data environment.

For adjusting the biometric authentication procedure with the searchable encryption scheme in a cloud computing environment and with big data capabilities, the following steps should be respected, as follows:

1. The user selects the biometric characteristic that they wish to use during the authentication process (e.g., fingerprint, facial recognition). Together with the identity and the acceptance of the user identity, a hash value is used within the cloud infrastructure and big data servers.
2. The two pieces of information (*user_digital_data* and *biometric_user_template*) are transmitted further to the *ComputationService*, which runs on a server placed within *Searching Cloud* (see Figure 6).
3. The *UserBiometricDataDBService* queries the database from the BUDB server for the biometric user template that is linked with the *user_digital_data*. The output is represented by *queryResult*.
4. Once the biometric user template is received, the *ComputationService* computes the distance based on the *queryResult* and *biometric_user_template* using the *computeDis-*

tance function. To compute the distance, we used Euclidean distance and Hamming distance. The reason we used both distances was to assess which one was faster on different simulation architectures. Theoretically speaking, we denote the distance as $\delta = distance\left(\widetilde{\phi}, \widetilde{\theta}\right)$, where ϕ is the user's digital data, and θ is the biometric user template. Note the fact that in *user_digital_data* we do not use the claimed identity that the user provided in the first phase; it is an important aspect of computing the distance because the biometric template field from the database does not include in its content the digital data. Digital data are a separate field.

The *AuthenticationService* receives the *distance* and proceeds with a derivation process between the *user_digital_data* and *biometric_template_record.* The derivation process is compared with the accuracy level. The accuracy level is set by the system or a dedicated server from the cloud infrastructure and designed in such a way that if the templates are close enough, the user is authenticated with success, otherwise, the user is not authenticated and is asked to repeat the authentication process.

8. Conclusions

This paper proposed a system that has three components: classical authentication, biometric authentication, and searchable encryption. The proposed system has applicability in many domains that work with and need to query data. We have seen that the submodules of the proposed system can be implemented as services; therefore, it is suitable for a wide range of applications. We have shown and demonstrated that the proposed scheme can be implemented with success for medium to complex network infrastructures. Our increased attention was dedicated to the trapdoor function, which, based on the generated value, led to the searching process being performed with success. The correctness that we computed for the operations guaranteed that the cloud servers would return the correct documents.

The main challenges experienced during the current research were related to the cloud environment. We pushed everything (in terms of processes, services, and testing applications) to the Microsoft Azure cloud platform, which gave us the most amazing experience due to the complexity of the authentication system.

In subsequent research, we will move our process to the Google Cloud Platform, where we will try the same authentication method, compare the results obtained with those from Microsoft Azure, and decide which platform is more reliable. The next direction is to adjust our authentication scheme for Internet-of-Things (IoT) devices.

In the end, the proposed contributions were achieved with success, which led us to positive results and unique contributions, which renders our scheme a real candidate and challenger for other authentication mechanisms. The unique characteristic of the authentication scheme is provided by the integration of searchable encryption.

Author Contributions: Conceptualization, M.I.M. and S.L.N.; Methodology, S.L.N.; Validation, S.L.N.; Formal Analysis, M.I.M.; Investigation, M.I.M. and S.L.N.; Resources, M.I.M. and S.L.N.; Writing – original draft preparation, M.I.M. and S.L.N.; Writing – review and editing, M.I.M.; Visualization, M.I.M.; Supervision, M.I.M.; Project administration, M.I.M. and S.L.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Appendix A

For our research, we considered the simulation of the following scenario in which we considered an HTTP request during the authentication process that would keep requesting and hitting at the same endpoint within the same domain. When simulating the case of a user inputting wrong credentials (username, password, and biometric characteristic, as per our research), the user is informed that his credentials are invalid and is prompted to try different credentials. If the process is repeated in a foolish manner or intentionally, the user is restricted, and additional attempts will be prohibited.

The below script (Figure A1) helped us to generate and simulate a repeat request, as follows:

```
{
"port": 56789,
"protocol": "http",
"name": "authentication malicious user",
"stubs": [
{
"predicates": [
{
"equals": {
"method": "post",
"path": "/users/login"
}
}
],
"responses": [
{
"is": {
"statusCode": 200,
"body": "The user has been successfully loged in."
}
},
{
"is": {
"statusCode": 400,
"body": "The credentials are wrong. You can try 2 more times."
}
},
{
"is": {
"statusCode": 400,
"body": "The credentials are wrong. You can try 1 more time."
}
},
{
"is": {
"statusCode": 400,
"body": "The credentials are wrong. You cannot try anymore. Your authentication attempts are finished."
}
}
]
}
]
}
```

The next step is represented by testing the repeating requests. The following has been tested using the C# programming language and it looks as follows:

```
using System;
using xUnit;

namespace MDPI_Cryptography_TestUserAuthentication
{
public class MDPI_UnitTest_UserAuthentication
{
MDPI_AuthenticateUser authenticationMDPIUser = new MDPI_AuthenticateUser();
```

Figure A1. *Cont.*

```
//The method that follows [Fact] attribute represents
      //a unit test according to xUnit testing framework.
      //More details: https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-
with-dotnet-test
[Fact]
public void LoginWithSuccess()
{
var supposedMessage = "correct credentials";
var generalMessage = "The user has been logged with success.";
var goodState = authenticationMDPIUser.Login(supposedMessage);

      //Assert it is a Class which represents a collection of helper
      //classes to test different conditions within the unit test.
      //If the condition that is tested is not satisfied, an exception is thrown.
      //More details: https://docs.microsoft.com/en-us/dotnet/api/microsoft.visualstu-
dio.testtools.unittesting.assert?view=visualstudiosdk-2022
Assert.Equal(generalMessage, goodState);
}

[Fact]
public void FaillingLoginProcess_First()
{
var supposedMessage = "wrong user credentials";
var generalMessage = "Incorrect login. You have 2 more attempts left.";
var goodState = authenticationMDPIUser.Login(supposedMessage);
Assert.Equal(generalMessage, goodState);
}

[Fact]
public void FaillingLoginProcess_Second()
{
var supposedMessage = "wrong user credentials";
var generalMessage = "Incorrect login. You have 1 more attempt left.";
var goodState = authenticationMDPIUser.Login(supposedMessage);
Assert.Equal(generalMessage, goodState);
}

[Fact]
public void FaillingLoginProcess_Third()
{
var supposedMessage = "wrong user credentials";
var generalMessage = "Incorrect login. You have no more attempts left.";
var goodState = authenticationMDPIUser.Login(supposedMessage);
Assert.Equal(generalMessage, goodState);
}
}
}
```

Figure A1. Generating and simulating repeated requests.

References

1. Ransomware Exploits and Supply Chain Attacks Lead the Cyber Trends in the First Half of 2021. Available online: https://pages.checkpoint.com/cyber-attack-2021-trends.html (accessed on 10 January 2022).
2. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
3. Bösch, C.; Hartel, P.; Jonker, W.; Peter, A. A Survey of Provably Secure Searchable Encryption. *ACM Comput. Surv.* **2014**, *47*, 1–51. [CrossRef]
4. Handa, R.; Krishna, C.R.; Aggarwal, N. Searchable encryption: A survey on privacy-preserving search schemes on encrypted outsourced data. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e5201. [CrossRef]
5. Jain, A.; Bolle, R.; Pankanti, S. (Eds.) *Biometrics: Personal Identification in Networked Society*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006; Volume 479.
6. Liu, S.; Silverman, M. A practical guide to biometric security technology. *IT Prof.* **2001**, *3*, 27–32. [CrossRef]
7. IEEE Biometric Databases. Available online: https://ieee-biometrics.org/index.php/resources/biometric-databases (accessed on 3 December 2021).
8. Biometric Dataset Collections. Available online: https://citer.clarkson.edu/research-resources/biometric-dataset-collections-2/ (accessed on 3 January 2022).
9. Marascu, V.; Stancu, C.; Satulu, V.; Bonciu, A.; Grisolia, C.; Dinescu, G. Material Erosion and Dust Formation during Tungsten Exposure to Hollow-Cathode and Microjet Discharges. *Appl. Sci.* **2020**, *10*, 6870. [CrossRef]

10. Marascu, V.; Lazea–Stoyanova, A.; Bonciu, A.; Satulu, V.; Dinescu, G. Tungsten particles fabrication by a microjet discharge. *Mater. Res. Express* **2020**, *7*, 066509. [CrossRef]
11. Marascu, V.; Lazea-Stoyanova, A.; Stancu, C.; Dinescu, G. The influence of plasma operation parameters on synthesis process of copper particles at atmospheric pressure. *Plasma Process. Polym.* **2017**, *15*, e1700091. [CrossRef]
12. Ryu, R.; Yeom, S.; Kim, S.-H.; Herbert, D. Continuous Multimodal Biometric Authentication Schemes: A Systematic Review. *IEEE Access* **2021**, *9*, 34541–34557. [CrossRef]
13. Sain, M.; Normurodov, O.; Hong, C.; Hui, K.L. A Survey on the Security in Cyber Physical System with Multi-Factor Authentication. In Proceedings of the 2021 23rd International Conference on Advanced Communication Technology (ICACT), Pyeongchang, Korea, 7–10 February 2021; pp. 1–8. [CrossRef]
14. Bezzateev, S.; Davydov, V.; Ometov, A. On Secret Sharing with Newton's Polynomial for Multi-Factor Authentication. *Cryptography* **2020**, *4*, 34. [CrossRef]
15. Khalid, H.; Hashim, S.; Ahmad, S.S.; Hashim, F.; Chaudhary, M. SELAMAT: A New Secure and Lightweight Multi-Factor Authentication Scheme for Cross-Platform Industrial IoT Systems. *Sensors* **2021**, *21*, 1428. [CrossRef]
16. Zhang, J.; Zhong, H.; Cui, J.; Xu, Y.; Liu, L. SMAKA: Secure Many-to-Many Authentication and Key Agreement Scheme for Vehicular Networks. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 1810–1824. [CrossRef]
17. Cheng, G.; Chen, Y.; Deng, S.; Gao, H.; Yin, J. A Blockchain-Based Mutual Authentication Scheme for Collaborative Edge Computing. *IEEE Trans. Comput. Soc. Syst.* **2021**, *9*, 146–158. [CrossRef]
18. Kim, D.; Seo, S.; Kim, H.; Lim, W.-G.; Lee, Y. A Study on the Concept of Using Efficient Lightweight Hash Chain to Improve Authentication in VMF Military Standard. *Appl. Sci.* **2020**, *10*, 8999. [CrossRef]
19. Lee, H.; Ryu, J.; Lee, Y.; Won, D. Security Analysis of Blockchain-based User Authentication for Smart Grid Edge Computing Infrastructure. In Proceedings of the 2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Korea, 4–6 January 2021; 2021; pp. 1–4.
20. Xia, X.; Ji, S. An Efficient Anonymous Authentication Scheme for Privacy-preserving in Smart Grid. In Proceedings of the 2021 IEEE Conference on Dependable and Secure Computing (DSC), Aizuwakamatsu, Japan, 30 January–2 February 2021; pp. 1–2. [CrossRef]
21. Laka, P.; Korzeb, Z.; Mazurczyk, W. Novel user authentication method based on body composition analysis. *Ann. Telecommun.* **2021**, *76*, 175–185. [CrossRef]
22. Ibrahim, D.R.; The, J.S.; Abdullah, R. Multifactor authentication system based on color visual cryptography, facial recognition, and dragonfly optimization. *Inf. Secur. J. A Glob. Perspect.* **2021**, *30*, 149–159. [CrossRef]
23. Wong, A.; Furukawa, M.; Maeda, T. Robustness of Rhythmic-Based Dynamic Hand Gesture with Surface Electromyography (sEMG) for Authentication. *Electronics* **2020**, *9*, 2143. [CrossRef]
24. Yu, Y.; Shi, J.; Li, H.; Li, Y.; Du, X.; Guizani, M. Key-Policy Attribute-Based Encryption With Keyword Search in Virtualized Environments. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1242–1251. [CrossRef]
25. Khader, D. Introduction to Attribute Based Searchable Encryption. In *Communications and Multimedia Security. CMS 2014. Lecture Notes in Computer Science*; De Decker, B., Zúquete, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8735. [CrossRef]
26. Li, J.; Wang, M.; Lu, Y.; Zhang, Y.; Wang, H. ABKS-SKGA: Attribute-based keyword search secure against keyword guessing attack. *Comput. Stand. Interfaces* **2021**, *74*, 103471. [CrossRef]
27. Wang, S.; Yao, L.; Zhang, Y. Attribute-based encryption scheme with multi-keyword search and supporting attribute revocation in cloud storage. *PLoS ONE* **2018**, *13*, e0205675. [CrossRef]
28. Yin, H.; Xiong, Y.; Zhang, J.; Ou, L.; Liao, S.; Qin, Z. A Key-Policy Searchable Attribute-Based Encryption Scheme for Efficient Keyword Search and Fine-Grained Access Control over Encrypted Data. *Electronics* **2019**, *8*, 265. [CrossRef]
29. Li, J.; Zhang, Y.; Ning, J.; Huang, X.; Poh, G.S.; Wang, D. Attribute Based Encryption with Privacy Protection and Accountability for CloudIoT. *IEEE Trans. Cloud Comput.* **2020**. [CrossRef]
30. Cui, Y.; Gao, F.; Shi, Y.; Yin, W.; Panaousis, E.; Liang, K. An Efficient Attribute-Based Multi-Keyword Search Scheme in Encrypted Keyword Generation. *IEEE Access* **2020**, *8*, 99024–99036. [CrossRef]
31. Chi, P.-W.; Wang, M.-W.; Shiu, H.-J. How to Hide the Real Receiver Under the Cover Receiver: CP-ABE With Policy Deniability. *Access IEEE* **2020**, *8*, 89866–89881. [CrossRef]
32. Lee, S.; Jo, H.J.; Choi, W.; Kim, H.; Park, J.H.; Lee, D.H. Fine-Grained Access Con-trol-Enabled Logging Method on ARM TrustZone. *Access IEEE* **2020**, *8*, 81348–81364. [CrossRef]
33. Zhang, L.; Su, J.; Mu, Y. Outsourcing Attributed-Based Ranked Searchable Encryption With Revocation for Cloud Storage. *IEEE Access* **2020**, *8*, 104344–104356. [CrossRef]
34. Zhang, K.; Liu, X.; Li, Y.; Zhang, T.; Yang, S. A Secure Enhanced Key-Policy Attribute-Based Temporary Keyword Search Scheme in the Cloud. *IEEE Access* **2020**, *8*, 127845–127855. [CrossRef]
35. Zhong, Y.; Ma, S.; Huang, Q. Plaintext-Verifiably-Checkable Encryption and Its Extension in Dual-Server Setting. *IEEE Access* **2020**, *8*, 132825–132840. [CrossRef]
36. Du, R.; Zhang, Y.; Li, M. Database Padding for Dynamic Symmetric Searchable Encryption. *Secur. Commun. Netw.* **2021**, *2021*, 9703969. [CrossRef]
37. Fan, K.; Chen, Q.; Su, R.; Zhang, K.; Wang, H.; Li, H.; Yang, Y. MSIAP: A Dynamic Searchable Encryption for Privacy-Protection on Smart Grid with Cloud-Edge-End. *IEEE Trans. Cloud Comput.* **2021**, 1. [CrossRef]

38. Zuo, C.; Lai, S.; Yuan, X.; Liu, J.K.; Shao, J.; Wang, H. Searchable Encryption for Conjunctive Queries with Extended Forward and Backward Privacy. Cryptology ePrint Archive. 2021. Available online: https://eprint.iacr.org/2021/1585. (accessed on 10 January 2022).
39. Zhu, X.; Fu, S.; Hu, H.; Wu, Q.; Liu, B. Efficient boolean SSE: A novel encrypted database (EDB) for biometric authentication. *Int. J. Intell. Syst.* **2021**, 1–19. [CrossRef]
40. Haghighat, M.; Zonouz, S.; Abdel-Mottaleb, M. Identification Using Encrypted Biometrics. In Proceedings of the International Conference on Computer Analysis of Images and Patterns, York, UK, 27–29 August 2013; Springer International Publishing: Berlin/Heidelberg, Germany, 2013; pp. 440–448.
41. Cachet, C.; Ahmad, S.; Demarest, L.; Hamlin, A.; Fuller, B. Proximity Searchable Encryption for the Iris Biometric. Cryptology ePrint Archive. 2020. Available online: https://eprint.iacr.org/2020/1174 (accessed on 5 January 2022).
42. Zhang, P.; Chui, Y.; Liu, H.; Yang, Z.; Wu, D.; Wang, R. Efficient and Privacy-Preserving Search over Edge-Cloud Collaborative Entity in IoT. *IEEE Internet Things J.* **2021**, 1. [CrossRef]
43. Gao, H.; Luo, S.; Ma, Z.; Yan, X.; Xu, Y. BFR-SE: A Blockchain-Based Fair and Reliable Searchable Encryption Scheme for IoT with Fine-Grained Access Control in Cloud Environment. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 1–21. [CrossRef]
44. Zhang, H.; Yang, Z.; Yu, H. Lightweight and Privacy-preserving Search over Encryption Blockchain. In Proceedings of the 2021 7th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC), Beijing, China, 17–19 November 2021; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA, 2021; pp. 423–427.
45. Stallings, W. *Cryptography and Network Security: Principles and Practice*; Prentice Hall: Hoboken, NJ, USA, 2019.
46. Acar, A.; Aksu, H.; Uluagac, A.S.; Conti, M. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv.* **2018**, *51*, 1–35. [CrossRef]
47. Dinesha, H.A.; Agrawal, V.K. Multi-level Authentication Technique for Accessing Cloud Services. In Proceedings of the International Conference on Computing, Communication and Applications (ICCCA), Dindigul, India, 22–24 February 2012; IEEE: New York, NY, USA, 2012; pp. 1–4.
48. Kapczyński, A.; Sobota, M. Distributed Authentication Systems Enhanced by Quantum Protocols. In Proceedings of the Fifth International Conference on Information Technology: New Generations (itng 2008), Washington, DC, USA, 2008, 7–9 April 2008; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA; pp. 928–931.
49. Srivastava, P.; Singh, S.; Pinto, A.A.; Verma, S.; Chaurasiya, V.K.; Gupta, R. An Architecture Based on Proactive model for Security in Cloud. In Proceedings of the International Conference on Recent Trends in IT, Chennai, India, 3–5 June 2011; IEEE: New York, NY, USA, 2011; pp. 661–666.
50. Choudhury, A.J.; Kumar, P.; Sain, M.; Lim, H.; Jae-Lee, H. A Strong User Authentication Framework for Cloud Computing. In Proceedings of the Asia-Pacific Services Computing Conference, Yilan, Taiwan, 9–12 December 2008; IEEE Computer Society: New York, NY, USA, 2011; pp. 110–115.
51. Revar, A.G.; Bhavsar, M.D. *Securing User Authentication Using Single Sign-On in Cloud Computing*; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA, 2011.
52. Kanjee, M.R.; Divi, K.; Liu, H. A Physiological Authentication Scheme in Secure Healthcare Sensor Networks. In Proceedings of the 2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), Boston, MA, USA, 21–25 June 2010.
53. Zhang, W. 2-Tier Cloud Architecture with Maximized RIA. *Res. Inst. Appl. Comput. Technol. IEEE* **2010**, *6*, 52–56.
54. Zhao, F.; Peng, X.; Zhao, W. Multi-Tier Security Feature Modeling for Service-Oriented Application Integration. In Proceedings of the 2009 Eighth IEEE/ACIS International Conference on Computer and Information Science, Shanghai, China, 1–3 June 2009; IEEE: New York, NY, USA, 2009; pp. 1178–1183.
55. Enisa Threat Landscape. 2021. Available online: http://tinyurl.com/5n7r3pjr (accessed on 12 January 2021).
56. Boneh, D.; Di Crescenzo, G.; Ostrovsky, R.; Persiano, G. Public key encryption with keyword search. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 506–522.
57. Byun, J.W.; Rhee, H.S.; Park, H.-A.; Lee, D.H. Off-Line Keyword Guessing Attacks on Recent Keyword Search Schemes over Encrypted Data. In *Selected Areas in Cryptography*; Springer International Publishing: Berlin/Heidelberg, Germany, 2006; pp. 75–83.
58. Bours, P.; Mondal, S. Performance evaluation of continuous authentication systems. *IET Biom.* **2015**, *4*, 220–226. [CrossRef]
59. Song, D.X.; Wagner, D.; Perrig, A. Practical techniques for searches on encrypted data. In Proceedings of the 2000 IEEE Symposium on Security and Privacy, SP2000, Berkeley, CA, USA, 14–17 May 2000; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA, 2000; pp. 44–55.