applied sciences

MDPI

Article

Blowfish Hybridized Weighted Attribute-Based Encryption for Secure and Efficient Data Collaboration in Cloud Computing

Smarajit Ghosh [1] and Vinod Karar [2,*]

[1] Department of Electrical and Instrumentation Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab-147004, India; smarajitg@hotmail.com
[2] Optical Devices and Systems, CSIR-Central Scientific Instruments Organization, Sector 30-C, Chandigarh-160030, India
* Correspondence: vkarar@rediffmail.com; Tel.: +91-987-881-5022

Received: 16 April 2018; Accepted: 30 June 2018; Published: 11 July 2018

Abstract: Cloud computing plays a major role in sharing data and resources to other devices through data outsourcing. During sharing resources, it is a challenging task to provide access control and secure write operations. The main issue is to provide secure read and write operations collaboratively and to reduce computational overload by effective key management. In this paper, a secure and an efficient data collaboration scheme blowfish hybridized weighted attribute-based Encryption (BH-WABE) for secure data writing and proficient access control has been proposed. Here, weight is assigned to each attribute based on its importance and data are encrypted using access control policies. The cloud service provider stores the outsourced data and an attribute authority revokes or updates the attributes by assigning different attributes based on the weight. The receiver can access the data file corresponding to its weight in order to reduce the computational overload. The proposed BH-WABE provides collusion resistance, multiauthority security and fine-grained access control in terms of security, reliability, and efficiency. The performance is compared with the conventional hybrid attribute-based encryption (HABE) scheme in terms of data confidentiality, flexible access control, data collaboration, full delegation, partial decryption, verification, and partial signing.

Keywords: cloud computing; secure write operation; data encryption; key management scheme; fine grained access control; multiauthority security; data collaboration; ABE; HIBE; HABE

1. Introduction

Attribute-based encryption (ABE) is a popular cryptographic technology to protect the security of users' data in cloud computing. Cloud computing is one of the biggest areas because of its high-level features such as convenience, scalability, and cost-saving. Due to its vulnerability, the development of the security model is very difficult. Consequently, the economic benefit and availability will be affected [1,2]. The attacker constructs the attacks in mobile application and devices in that place develop the hypervisor to destroy the virtual machine (VM) side-channel attack and denial-of-service (DOS) attack. Cloud computing will be affected by the presence of traffic, in which case IP addresses are used to eliminate the traffic [3]. The data collaboration service, as a promising service offered by the cloud service provider (CSP), is to support the availability and consistency of the shared data among users.

In cloud storage devices, data are stored among multiple users, not only in the cloud. A technique like privacy-preserving is used to allow public auditing through this way. The data will be stored, the integrity of shared data will be checked, and then the information will be verified by the ring

structures. The shared data have a number of blocks containing the signer identity and the information are kept secret from third parties until the verification of shared information [4].

An organization has owners, and users can store data in the cloud and check the data for security purposes. In cloud computing systems, strong security obstacles and privacy issues are adapted in which related terms like confidentiality, integrity, control, audit, and availability are provided to secure the data [5]. In order to secure the stored data by the decentralized access control scheme in which the stored information can be decrypted by the valid users and the key distribution process by manner of decentralized way. All files or records will be stored in the cloud by access policy, which is known by the cloud [6]. The network security attackers are "viruses, trojan horses, man in the middle attacks, back doors, denials of service" and so on. In order to obtain flexibility and economic savings, the local sites are transformed to the commercial public cloud and it will be motivated by the data owners to outsource the complex data [7,8].

Before storing the information on the cloud, the cloud checks the authenticity of the user without the estimation of the user's identity. The valid users can decrypt the stored data and then support modification, creation, and reading the data due to the prevention of replay attacks [9,10]. The cloud computing environments assure data confidentiality and access control methods to utilize key policy ABE and proxy re-encryption. To avoid collision attacks in this method, the data file into header and body is divided [11,12]. The data are to be stored in the cloud, where traditional security is not well suited. The cloud data will be more secure from duplication. The static and dynamic-based tree structures are constructed here for securing the data. The random elements are collected from the client by static tree structures [13].

By the use of cryptographic secret-sharing, the secured data are transmitted across multiple clouds from which the data can be accessed by attribute-based encryption. The issues like hacking threats, internally or externally, infeasibility of encrypting data. For this purpose, the user is permitted for data encryption by employing key-based techniques to provide confidentiality [14,15]. For security purposes, the digital signature with an Rivest–Shamir–Adleman (RSA) algorithm has been explored to the cloud data and it authenticates the digital message. The anonymity of the user is guaranteed by the combined scheme of attribute based signature (ABS) and group signature in which the private key protects attribute authorities [16,17]. The encrypted data are increased to introduce a large number of keywords and the trapdoor generation algorithm is used to solve the out of order problem without loss of data [18].

The personal data are kept secret in the cloud to protect sensitive data and remove the constraints. The data owner encrypts the data that will be outsourced to the cloud by the fine grained-access control. Between users and the cloud, the information will be leaked during the collusion time and it can be avoided by the safety data sheet (SDS) frame [19,20]. The similarity index is proposed to protect the data from insecurity and the m-index is encrypted to support the neighbor's queries. The key policy attribute based signature (KP-ABS) scheme composes the signer's private key into two components. The other users cannot forge the signature [21,22].

In this proposed work, blowfish hybridized weighted attribute-based encryption (BH-WABE) algorithm is developed for securing the data stored in the cloud. An attribute encryption scheme with more authority is more suitable for data access control cloud storage systems because the user can be held by multiple institutions to manage property, and access to policy data owners to use the property that may be defined in different institutions. Traditional single authority to manage all user attributes dense steel, easy to degrade system performance. In addition, a single authority solution requires a completely honest authorized body; it is difficult to meet the security requirements of cloud computing environments. Weighted attribute-based encryption is hybridized with the blowfish algorithm for encryption purposes. Encryption, key generation, and decryption are ensured with the blowfish algorithm. A key contribution in this paper is summarized as follows:

- We propose a novel data collaboration scheme for secure read and write operations in cloud computing that allows a symmetric encryption algorithm for effective key management to reduce

computational overhead. A full delegation approach-based hybridized encryption (BH-WABE) that is employed for the outsourced data should be secure.

- ➢ We provide a verification method for the outsourced encryption and decryption. If the cloud returns incorrect results, users can notice it immediately by running the corresponding verification algorithm. Therefore, the user can access the data anywhere and anytime using any device. The computational cost is low, which is introduced by ABE in the user side.
- ➢ We provide a security and performance analysis of our scheme, which shows that our scheme is both secure and highly efficient.

2. Related Works

Some of the recent related works implemented in the recent past are discussed below:

Mobile devices, such as smartphones, which have been widely used by people to upload and download files, such as audio and video, also limit the sources in mobile devices. The cloud collects the files but the server does not have an idea about cloud security. Between users and the cloud, the data have been secured by classical access and provided lightweight security when the mobile accessing capacity of users became low. The watermarking scheme was developed by Wang et al. [23] in order to secure data between the cloud and users by authentication. The transmission errors could be minimized by combination of Reed–Solomon code with water marking.

In cryptographic techniques, the check ability was important and the versatility of access control had been enlarged by ABE method proposed by Li et al. [24]. The computational complexity, key issues, and decryption process were high in ABE method due to its high expensiveness. The constant efficiency was obtained by the user- and authority-side. To get the clear solution, the computing task had to send the third party and address the verifiable results by the third party.

The necessary resources like authentication and access control for computation of cloud control and integration management. The practical solutions were not suggested by role-based access control (RBAC) and context aware RBAC to the clients, which was based on dynamic access control. The new model, ontology based access model control (Onto-ACM), was used to address the limitation of cloud computing suggested by Choi et al. [25].

A process such as resource virtualization, global replication, and migration assured quality of service by the computing paradigm. The cloud storage data had cloud users hopeful, but the clear computing results were not obtained. The computation auditing secure protocol was proposed by Wei et al. [26] to secure storage and the process was completed with the batch verification, the signature verified by the designator, and sampling technique through this size was optimized and cost was minimized. The effectiveness and efficiency were clearly obtained from the experimental results.

The novel patient-centric framework had been proposed by Li et al. [27] to store personal records and access the data. The personal health record (PHR) files of each patient had been encrypted. Through this, clear and scalable data had been obtained, but it will be differed from the outsourcing of secure data by attribute-based encryption techniques. The multiple security domains degrade the complexity of key management due to the PHR system division by the scenario of multiple data. The security, scalability, and efficiency were enabled by break glass and access policy.

Subashini and Kavitha [28] presented a detailed survey regarding security issues in service delivery models in cloud computing and they discussed each method, along with their pros and cons.

3. Hierarchical Attribute-Based Encryption

By merging the features of ciphertext-policy-attribute-based encryption (CP-ABE) and hierarchical identity-based encryption (HIBE), one can derive hierarchical attribute-based encryption (HABE). Further, this scheme deals for fine-grained access control and scalability and also achieves full delegation by yielding key delegation between attribute authorities. Compared to the conventional schemes, this scheme symbolically represents the hierarchical structure of the enterprise, which is more appropriate to the environment of an organization outsourcing data in a cloud.

CP-ABE: It is an inverted model of key policy-attribute-based encryption (KP-ABE) that enables the data holder to explain the access strategy over the whole attributes that the data consumer wants to retain with the intention of decrypting the ciphertext. By doing so, confidentiality and data access control can be assured.

The CP-ABE algorithm involves four steps and it is represented below.

(1) **Setup ()**: This is a randomized part and it accepts only the unstated security parameter. Consequently, it yields the public key P_K and the master key M_K.
(2) **Encrypt** (P_K, S_a, m): This step fetches P_K, a message m and the descriptive attribute S_a as input. It outputs a cipher text C_T. 3).
(3) **Keygen** (M_K, AS): This step takes M_K and non-monotonic access structure AS as input and provides attribute secret key S_K for users as output.
(4) **Decrypt** (C_T, S_K): The input in this step is cipher text C_T, which contains the access tree T and the user's secret key S_K that is related to their descriptive attribute S_a, and the output is message m. This step is completed only if S_a satisfies T.

The access structure of CP-ABE is attached with the cipher text until the key for decryption process is interpreted with the pack of descriptive attributes as shown above. Consequently, the responsibility of KP-ABE is to change the characters of the cipher text and the decryption key. Furthermore, in this system, encryption provides the monotonic access form along with a threshold value for appropriate attributes. However, when the decryption key attributes fulfill the access policy in a known ciphertext, then only the ciphertext can be decrypted with the key. This method is more enthusiastic though the trusted server is negotiated. Generally, the CP-ABE approach is greater than the KP-ABE in terms of imposing encrypted data's access control. The major constraints of CP-ABE are that it cannot fulfill the necessities of initiatives in their access control as it requires efficiency and flexibility.

HIBE: The hybrid identity-based encryption (HIBE) is extended from IBE. Here, the private key is delivered by a solo private key generator (PKG) with the public keys as their primitive ID (PID), so-called as 1-HIBE in an overall identity-based encryption scheme and carry a drawback like heavy key managing. Therefore, to overcome this, a 2-HIBE scheme via a detailed definition of security is introduced that consists of domain PKG and a root PKG. The consumers and these are connected with a random string of PID. However, the domain PKG produces the private key to provide the requested domain secret key, which is acquired from the root PKG. Moreover, a root certificate authority (trustworthy third party) is involved by the cryptosystem, which permits a hierarchy of certificates. Through several levels of HIBE, the allotment of key escrow and root server workloads can be diminished.

HABE: The HABE algorithm, which combines the CP-ABE and HIBE, consists of the following five steps as given below.

HABE Algorithm	
Setup $(K) \geq P_K, M_{K0}$	// security parameter K as input, private key P_K and central authority's master key M_{K0} as output
Delegate $(P_K, M_{Kl}, S_a) \geq M_{Kl+1}$	// private key P_K, domain authority's master key M_{Kl} for a set of attributes S'_a, a set of attributes S_a where $S_a \in S'_a$, master key M_{Kl+1} of domain authority as output
KeyGen $(P_K, M_{Kl}, S_a) \geq S_K$	// private key P_K, domain authority's master key M_{Kl} and S_a set of attributes as input, attribute secret key S_K as output.
Encrypt $(P_K, m, T) \geq C_T$	// private key P_K, a message m and access policy tree T as input, cipher text C_T as the outcome.
Decrypt $(T, C_T, S_K) \geq m$, if $S_a \in T \perp otherwise$	// cipher text C_T, access tree policy T, attribute secret key S_K as input, a message m as output

Though various domain owners can manage the identical attribute, it is complicated to execute in practice. Furthermore, HABE cannot proficiently aid compound attributes and it may degrade the support for multivalue tasks. Therefore, to overcome these limitations, a new BH-WABE approach is proposed in this paper.

4. Proposed Blowfish Hybridized Weight Attribute-Based Encryption

In cloud computing, a secure and efficient data collaboration is achieved by the proposed BH-WABE approach. Most of the conventional ABE methods only have a single authority to handle both the secret and public keys. However, in many circumstances, the consumers hold attributes from multiauthority, and the data holders share data with consumers who are managed by a distinct authority. Many different multiauthority attribute-based access control structures have been developed to solve this problem. In access control systems with the intention of updating the ciphertext, a data holder has presented online for all time, besides the attributes that are given similar status. In the proposed scheme, the weighing of attributes is given by the blowfish algorithm to provide secure data in cloud computing.

The system involved five basic things: (a) the data holder, who encodes the data before uploading the data to the cloud under an access control policy; (b) a cloud server who provides data storing; (c) a weight attribute authority (WAA) to authorize, update and validate the attributes of users that are assigning different weights with respect to their prominence; (d) a Central Authority (CA), which allocates a global user identifier for each consumer as well as allots user public key to the WAA; and (e) the data consumers, as illustrated in Figure 1. In the proposed system, a blowfish algorithm is hybridized with weighted attributed authority as illustrated in Figure 1.

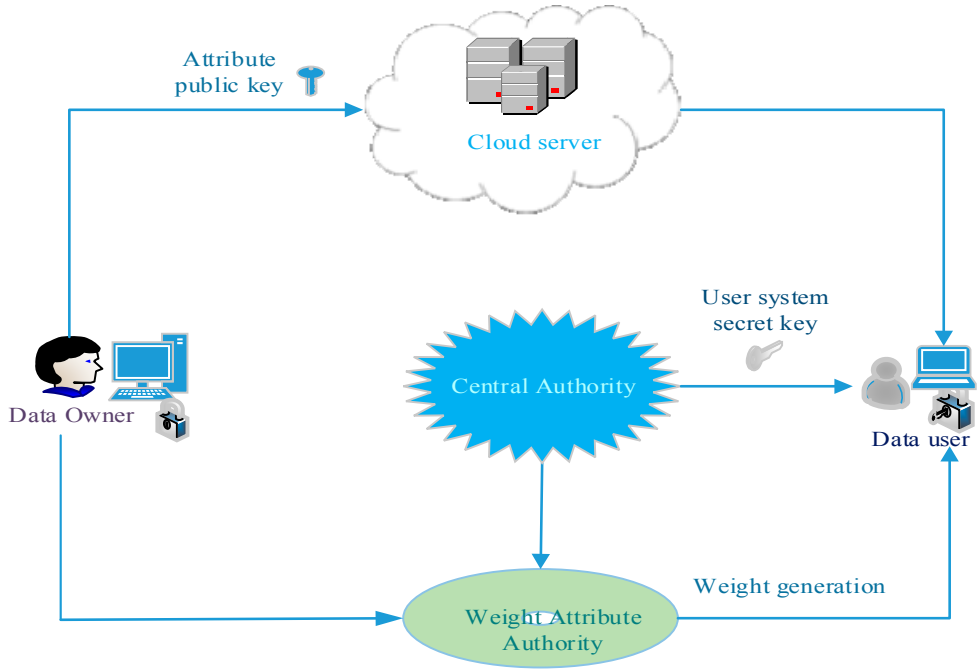

Figure 1. Proposed blowfish hybridized weighted attribute-based Encryption (BH-WABE) scheme.

In the proposed BH-WABE system model, the blowfish algorithm is applied to encrypt and decrypt data and to generate keys randomly. Moreover, an image-matching technique is employed for security purposes. Subsequently, the system generates weight value for users based on its attributes. For example, if User A = Dhoni from the HR department and User B = Sachin from the R&D department, both users initially encounter the security phase. Assuming that the system acknowledges that User A is valid, then the system generates weight values for User A based on its

attributes. According to the weight value, User A can decrypt the document, which is assigned to its corresponding weight. In contrast, User B cannot decrypt the document of User A. Though User B is a valid user, their weight rate does not match the weight rate of User A, but User B can decrypt its corresponding document based on its weight value. This approach is more prominent, reliable, and secure; besides, it is more applicable for real-time applications than the conventional methods in a cloud computing environment. BH-WABE encryption deals fine-grained access control, multiauthority security, and collusion resistance. The proposed scheme is represented in two phases: the algorithm phase and the system phase. At the algorithm phase, the blowfish algorithm is described along with system-level operations. Conversely, at system level, the high-level operations such as System Setup, User Annulment, New File Creation, New User admit, File Access and Deletion are explained.

4.1. Algorithm Level Operations

Blowfish Algorithm

Blowfish is a symmetric encryption algorithm [29]. It consists of a single key that is used for both encryption and decryption process. This blowfish encryption scheme's secret key ranges from 32 to 448 bits. If the range of key is 448 bits, then it needs 2448 groupings to define all the entire keys. Furthermore, this key has a fixed 64-bit block size with variable-length key block cipher. The cipher is a 16-round Feistel network, which uses password-dependent S-boxes to develop the structure by which the encryption and decryption process has taken place. This cipher divides messages into 64 bits blocks and then encrypts them separately.

The algorithm possesses two main sub-key groups, namely, the 18-entry P-boxes (permutation boxes) to perform bit-shuffling and four 256-entry S-boxes (substitution boxes) to perform simple nonlinear functions. Here, the S-boxes receive 8-bit as input and yield 32-bit output. The working principle of a single blowfish round is shown in Figure 2. The function F is the Feistel Function of Blowfish that splits half the 32-bit block in 8-bit chunks (quarters) and employs this quarter as input to the S-box. Subsequently, the outcomes of S-boxes are supplemented with the dropped carry, consequential in MOD 232 addition, and finally XOR operation has been performed. Conversely, the decryption process has been carried out by reversing the blowfish algorithm and is simply done by inverting P_{17} and P_{18} cipher blocks as well as by employing the P-entries in reverse order. Blowfish algorithm is generally divided into two sections, namely key-expansion and data encryption.

Key-expansion: In the Key expansion part, a 448-bit key is converted into numerous sub-key groups of 4168 bytes in aggregate. Normally, P-array is composed of 18 and 32-bit sub-keys ($P_1, P_2, \ldots\ldots P_{18}$) and four 32-bit S-Boxes, each containing 256 entries.

The procedures that involved in the key expansion process are given as follows:

Step 1: Set and Initialize S-box and P-box with values from the hexadecimal numbers of pi (<initial 3)

Step 2: The variable length of the user input key is XOR^{ed} with the P-entries until the entire P-array has been XOR^{ed} with input key bits.

Step 3: A block of zeroes is encrypted; subsequently the results are applied for P_1 and P_2 entries.

Step 4: Again, encrypt the ciphertext obtained from the encrypted zero block, then utilize for P_3 and P_4.

Step 5: Continue the process till each P-box entry and the S-box entry have been exchanged thus; totally, 521 iterations are necessary to generate all the essential sub-keys (i.e., 521 key generations).

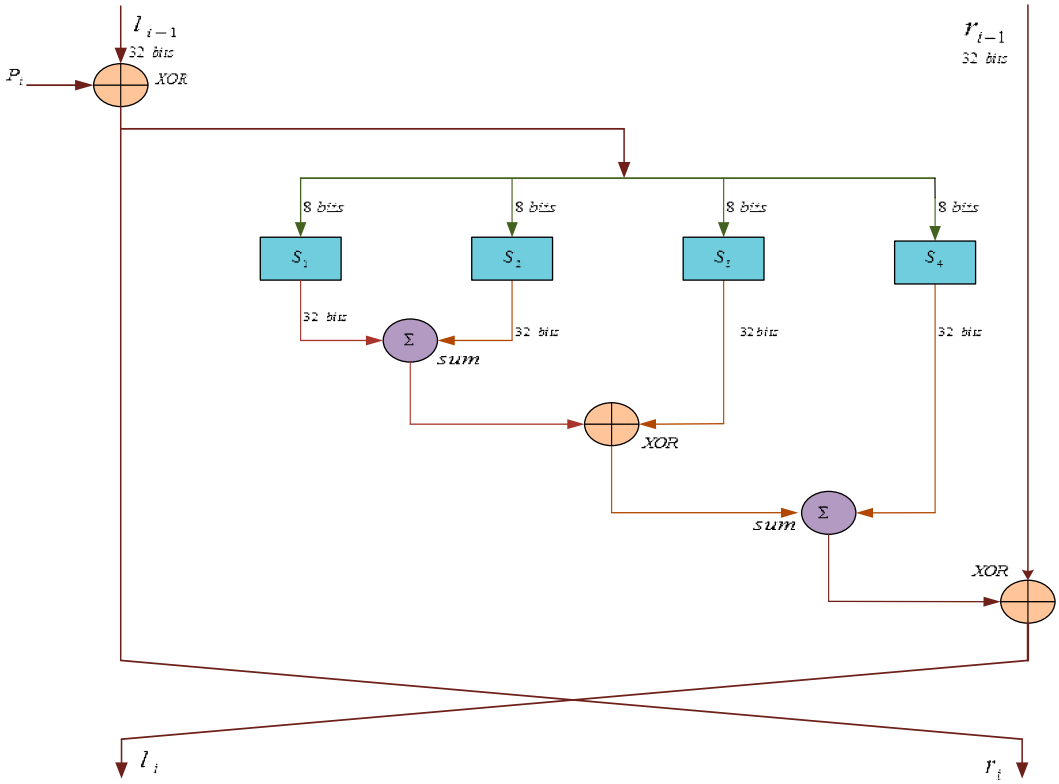

Figure 2. Single blowfish round.

Data-Encryption: Data encryption is carried out through 16 rounds. Further, each round performs key-based permutation and a key- and data-based replacement using XORs. Moreover, the addition operation (four indexed array data lookup tables) is performed on 32-bit words. Blowfish sums up all these features into a proficient and a dominant algorithm and it is illustrated below.

Blowfish Encryption Algorithm
Input: Y
Output: Recombined Y
Splits Y into 2, 32-bit halves: Y_l,Y_r
For i = 1 to 16
$Y_l = y_l \; XOR \; P_i$
$Y_r = F(y_l) \; XOR \; Y_r$
Swap y_l and Y_r
Swap y_l and Y_r (Undo the last swap)
$Y_r = Y_r \; XOR \; P_{17}$
$Y_l = Y_l \; XOR \; P_{18}$
Recombine Y_l and Y_r
End

The high optimisms and large assurances afforded by the blowfish algorithm produced much activity regarding a potential cryptanalysis.

4.2. System Level Process

System level processes in the proposed system are described as:

(1) **System Setup**: The challenger runs the global setup algorithm to obtain the global public parameters. The data holder selects a security parameter, subsequently sends a request to

the algorithm phase interface setup, as a consequence it yields the secret key S_K. The data holder then ciphers each S_K component and sends the encrypted components along with the signature to the central authority (CA). However, the CA authenticates the holder's signature. Further, if it is correct, the CA will utilize the system public and master key to create a public and secret key for a new user. Further, the WAA defines the weight for the attributes in the organization domain.

(2) **Key Generation**: When a new user wants to connect to the system, the CA will allocate a unique user ID to the consumer. However, the consumer then cyphers its attribute set, and sends it along with its signature to WAA. The attribute authority authenticates the consumer's signature. If it is right, WAA creates the equivalent attribute secret keys and weight for the new consumer. Subsequently, the CA and WAA, individually, convey the consumer's system secret key and attribute secret key to the new consumer. The challenger obtains the corresponding keys by running the authority setup and central authority setup algorithms and gives the public keys to the attackers.

(3) **Encryption**: Before uploading a data file to the cloud, the data holder initially logs in with a unique ID, and then randomly chooses a symmetric data file encryption key to encode the data. Furthermore, the data holder defines a "weighted threshold access structure" (W) for the corresponding data users and the data files and then encrypts the data with W as shown in Figure 3.

(4) **Decryption**: The data consumer initially downloads the data from the cloud to the local, and then requests the decryption algorithm to decrypt the data. If the data consumer's attributed secret key is authorized, then the system assigns dissimilar weights to different attributes as per their importance. Subsequently, the user can decrypt the corresponding data file with respect to W. Once a user is invalidated, then the user will not be able to decrypt the data file as shown in Figure 4.

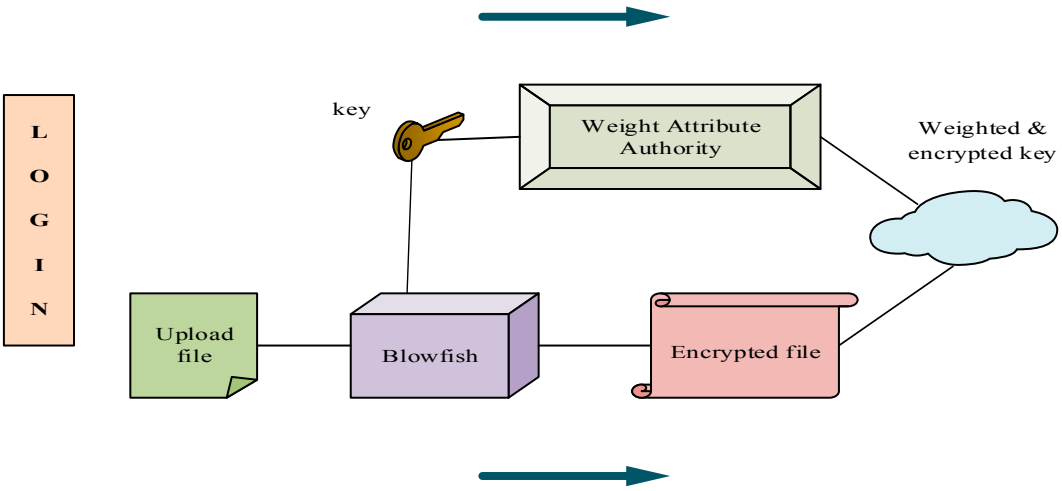

Figure 3. Data owner side (encryption process).

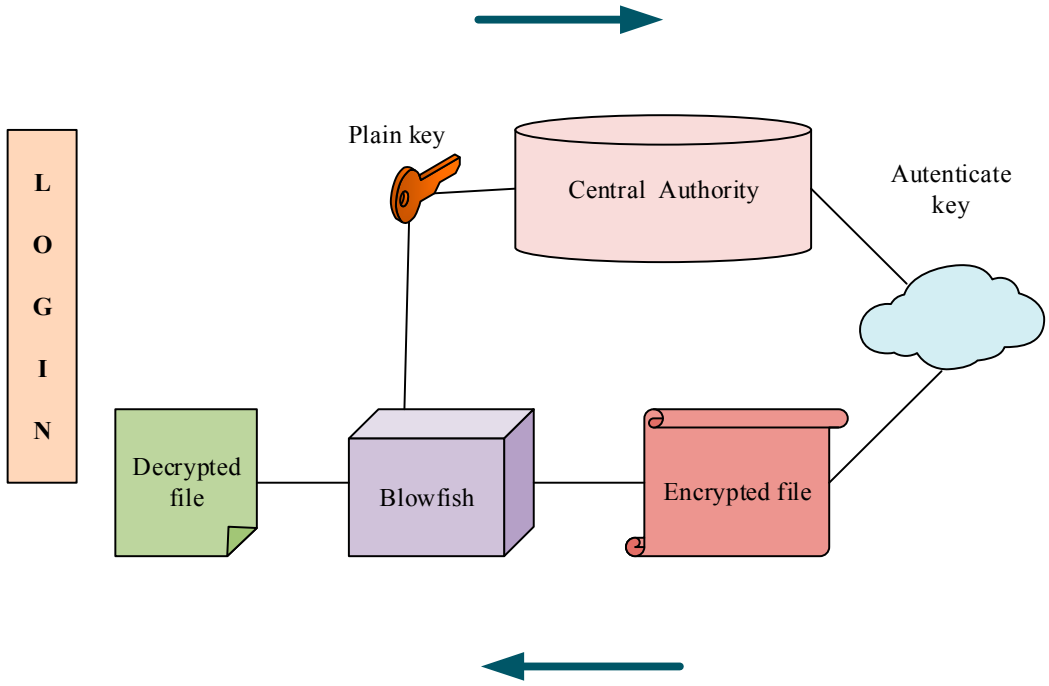

Figure 4. Data user side (decryption process).

5. Simulation Results and Discussion

In this section, the experiments are conducted for performance evaluation. The proposed scheme is implemented on the Java platform, in an Intel core with 2 GB RAM. The computation cost of encryption and decryption is computed. All of these systems not only provide data security, but also accomplish the access control of encrypted data on a cloud network. While comparing the data collaboration schemes of ABE [27] and HABE [30], the proposed BH-WABE attains partial signing, and full delegation, with less workload (data user and WAA) and also accomplishes lightweight key management in a large-scale consumer.

In the proposed scheme, various input files of different sizes (in kB) are encrypted and decrypted by blowfish algorithm. Key generation and weight generation are also done by blowfish algorithm. This algorithm is generated for security purpose and also it yields less execution timings for "encryption and decryption process". The security aspect of blowfish encryption approach has been enhanced. The final outcome of the proposed scheme is illustrated in Table 1. The time taken for encryption and decryption process by the proposed BH-WABE is compared with the conventional HABE scheme by considering the performance metrics.

Table 1. Experimental results of execution time of encryption/decryption, throughput for BH-WABE.

Input Data	File Size (GB)	Encryption Time (s)	Decryption Time (s)	Throughput
I_1	1	118	113	0.00847
I_2	2	230	221	0.00869
I_3	3	341	598	0.00874

Encryption time: The time taken to encrypt the data is termed as encryption time. It is used to estimate the throughput of an encryption approach as well as to evaluate the speed of the system. The encryption time can be determined by defining the time taken to generate a ciphertext from a plaintext and it is graphically represented in Figure 5.

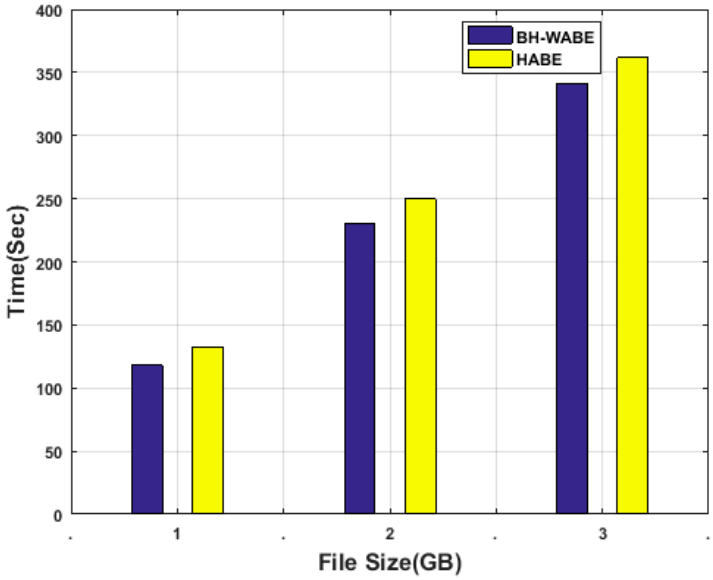

Figure 5. Computation time of encryption. HABE: hierarchical attribute-based encryption.

Decryption time: The inverse process of encryption process is termed as decryption process. However, the decryption time is defined as the time that a decryption algorithm takes to yield a plaintext from a ciphertext. The decryption time for the conventional HABE and the proposed BH-WABE is analyzed in Figure 6.

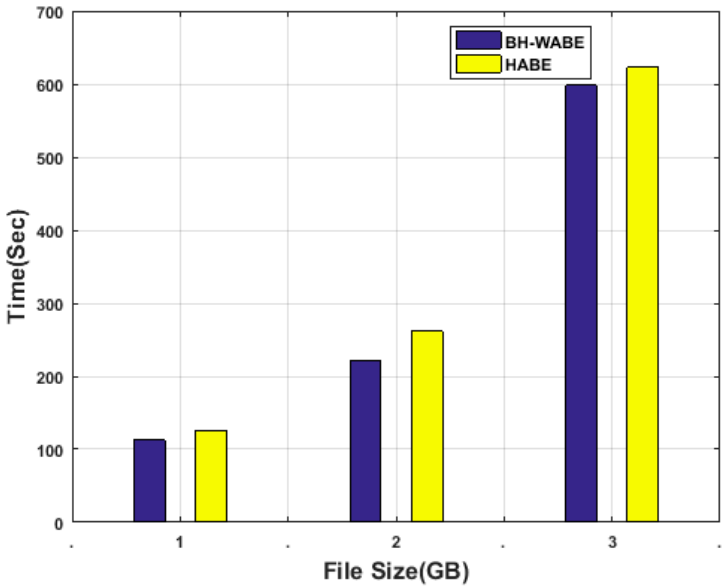

Figure 6. Computation cost of decryption.

Throughput: The ratio of the encrypted data file to the encryption time is referred as throughput. In Figure 7 it is seen that the throughput estimated from the proposed BH-WABE scheme is high when compared to the conventional HABE scheme.

$$\text{Throughput} = \frac{\text{File size (kB)}}{\text{Encryption time}} \quad (1)$$

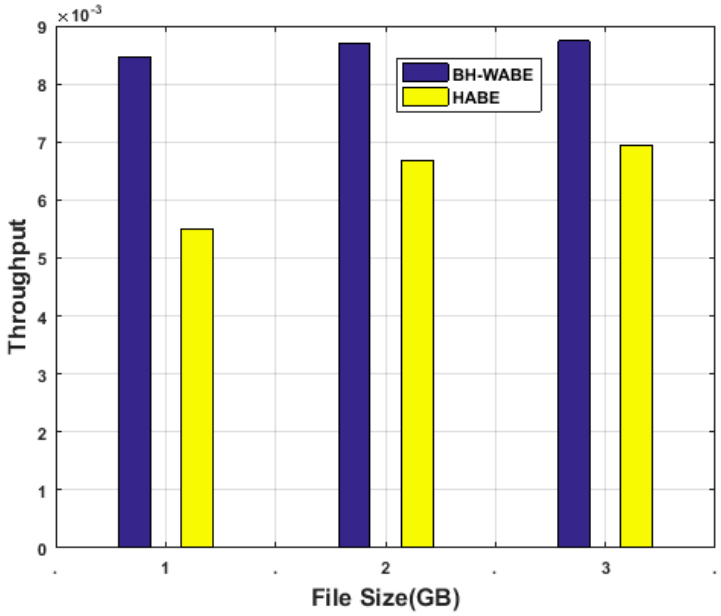

Figure 7. Comparison on throughput of HABE and BH-WABE.

Cost of secret key analysis: The computation cost and storage overhead of secret key are analyzed as plotted in Figure 8. The *y*-axis denotes storage overhead or time cost of a user's secret key. The *x*-axis denotes the number of weighted attributes in a user's secret key.

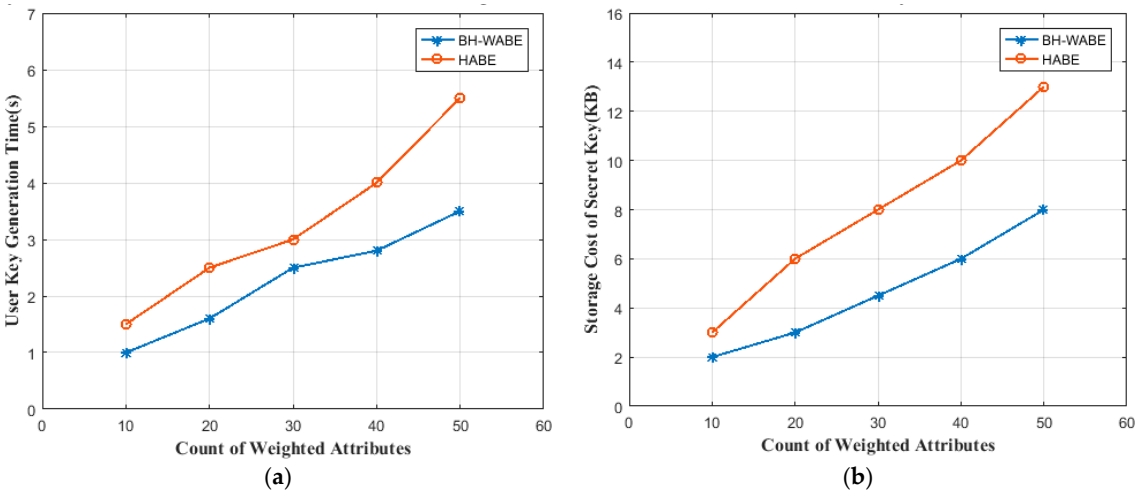

Figure 8. Cost of user secret key generation. (**a**) Time cost analysis; (**b**) Storage cost analysis.

Cost of ciphertext analysis: The storage overhead and computation cost of encrypting data by a data owner are compared as plotted in Figure 9. The *y*-axis denotes storage overhead or time cost of encrypting data by the data owner. The *x*-axis denotes the number of weighted attributes in access policy defined by the data owner.

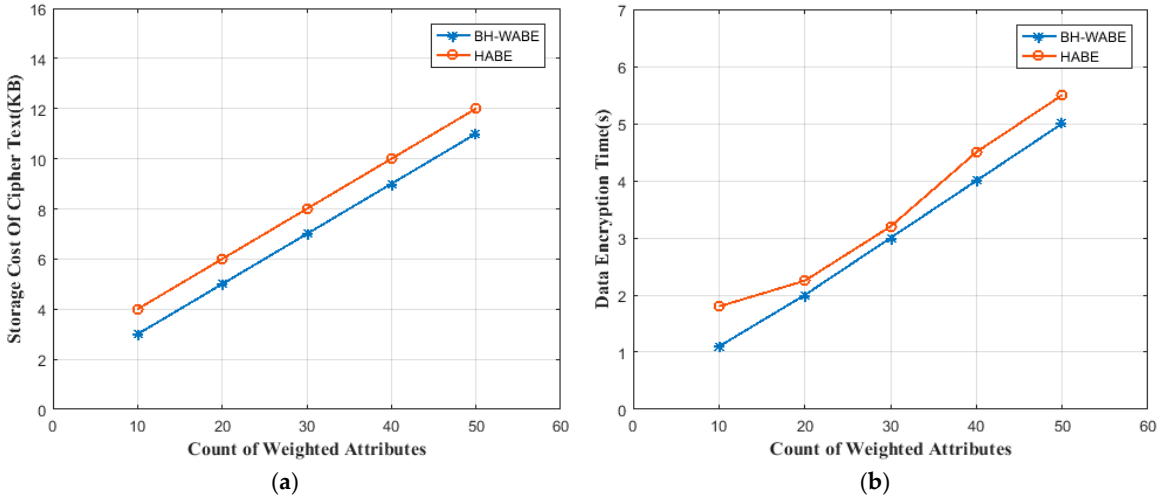

Figure 9. Cost of cipher text. (**a**) Storage cost analysis; (**b**) Time cost analysis.

5.1. Security Analysis

The sharing data in our scheme is encrypted with the BH-WABE technique, which is secure against a chosen plaintext attack if the decisional bilinear Diffie-Hellman (DBDH) assumption holds. We analyze the security properties of our scheme as follows:

Data confidentiality: The data is encrypted using access policy, and the confidentiality of data can be guaranteed against users who do not hold a set of attributes that satisfy the access policy. In the encryption phase, though the cloud performs encryption computation for the data owner, it still cannot access the data without the attribute key. During the decryption phase, since the set of attributes cannot satisfy the access policy in the ciphertext, the cloud server cannot recover the value $A = e(h,h)^{\gamma t}$ to further get the desired value of the global key (GK). Therefore, only the users with valid attributes that satisfy the access policy can decrypt the ciphertext. The data are encrypted with a random symmetric key CK, and then CK is protected by BH-WABE. Since the symmetric encryption and BH-WABE scheme are secure, the confidentiality of outsourced data can be guaranteed against unauthorized users whose identities are not in the set of receivers' identities defined by the data owner.

Fine-Grained Access control: The fine-grained access control allows flexibility in specifying differential access policies of individual data. To enforce this kind of access control, we utilize BH-WABE to escort the symmetric data encryption key. In the data encryption phase of our scheme, the data owner is able to enforce an expressive and flexible access policy and encrypt the symmetric key that is used to encrypt the data. Specifically, the access policy of encrypted data defined in the access tree supports complex operations, including both AND and OR gate, which is able to represent any desired access conditions.

5.2. Collusion Attack

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 so as to produce a new copy. In a 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 using the BH-WABE algorithm.

Without generality loss, assume J = {1,2}. Then, based on the encryption algorithm, the encrypted texts 'D' associated with policy $(N_1, \rho_1) \cap (N_2, \rho_2)$ is computed as follows:

$$D_0 = n \cdot \prod_{i=1,2} e(g,g)^{\alpha_i s_i} = ne(g,g)^{\alpha_1 s_1 + \alpha_2 s_2} \quad (2)$$

$$X_i = \left\{ g^{si}, Y_i = f^{si}, E_i = B_i^{si} \right\}_{j=1,2} \tag{3}$$

$$D_{i,j} = \left\{ \left(g^{x_i \lambda_{i,j}} z_{\rho_i(j)}^{-r_{i,j}}, C_{i,j} = g^{r_{i,j}} \right)_{j=1,\ldots l_i} \right\}_{i=1,2} \tag{4}$$

Let us consider two users, P1 and P2, with identifiers p1 and p2 respectively. In addition, user P1 owns attribute set S1, whose elements are monitored by authority A1, and S1 only satisfies the access structure (N_1, ρ_1) but not (N_2, ρ_2). Likewise, the other user, P2, owns attribute set S2, whose elements are monitored by authority A2, and S2 only satisfies the access structure (N_2, ρ_2) but not (N_1, ρ_1). Both users can get their keys from A1 and A2. However, if a user is not in the encryption list, then he cannot obtain (N_2, ρ_2) in the ciphertext. So, the decryption algorithm will fail. Specifically, only those intended recipients can decrypt this data. Therefore, users can only access the data they are allowed to and not the data they are not authorized to.

6. Conclusions

In a cloud environment, user authentication and data security are the challenging issues. Therefore, an efficient and scalable access control scheme has been proposed in this paper. Further, this scheme employs a blowfish hybridized weight attribute-based encryption mechanism not only to provide data security against the semi-trusted cloud service provider, but also the weight attribute authority and the central authority provides lightweight key management in large scale-consumers. Besides, the partial signing construction implemented in this scheme can reduce the computation overhead of user to the cloud server, which is efficient and appropriate for resource-constrained devices. Here, blowfish encryption and decryption algorithms are used to transmit data securely. When the authenticated user makes a request to the cloud, the corresponding files are sent to the consumer in an encrypted format based on its weight. Subsequently, the data consumer can decrypt the data using the key generated by the blowfish algorithm. The result shows that the proposed method BH-WABE is efficient in terms of security, reliability, and efficiency, as well as performing well when it is juxtaposed to the conventional HABE scheme by means of data confidentiality, flexible access control, data collaboration, full delegation, partial decryption, verification, and partial signing.

The future extent of the proposed work can be accessible, quality-based encryption and protection-saving property-based information-sharing with re-encryption. These are territories in which we can look into going ahead to use diverse methods to accomplish information-sharing.

Author Contributions: Both the authors have carried out the entire research work jointly.

Funding: The funding was provided by CSIR-Central Scientific Instruments Organization; Sector 30-C, Chandigarh-160030.

Acknowledgments: Authors are thankful to the Director of Thapar Institute of Engineering and Technology, Patiala, Punjab, India and Director, CSIR-Central Scientific Instruments Organization; Sector 30-C, Chandigarh-160030, India for providing the environment to carry out this work.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Modi, C.; Patel, D.; Borisaniya, B.; Patel, A.; Rajarajan, M. A survey on security issues and solutions at different layers of Cloud computing. *J. Supercomput.* **2013**, *63*, 561–592. [CrossRef]
2. Singh, S.; Young-Sik, J.; Park, J.H. A survey on cloud computing security: Issues, threats, and solutions. *J. Netw. Comput. Appl.* **2016**, *75*, 200–222. [CrossRef]
3. Yan, Q.; Yu, F.R. Distributed denial of service attacks in software-defined networking with cloud computing. *IEEE Commun. Mag.* **2015**, *53*, 52–59. [CrossRef]
4. Ranchal, R.; Bhargava, B.; Othmane, L.B.; Lilien, L.; Kim, A.; Kang, M. An Approach for Preserving Privacy and Protecting Personally Identifiable Information in Cloudcomputing. Available online: https://www.cs.purdue.edu/homes/bb/cs590/handouts/PII_Cloud.pdf (accessed on 16 April 2018).

5. Zhou, M.; Zhang, R.; Xie, W.; Qian, W.; Zhou, A. Security and privacy in cloud computing: A Survey. In Proceedings of the Sixth International Conference on Semantics Knowledge and Grid (SKG), Washington, DC, USA, 1–3 November 2010; pp. 105–112. [CrossRef]
6. Ruj, S.; Stojmenovic, M.; Nayak, A. Decentralized access control with anonymous authentication of data stored in clouds. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 384–394. [CrossRef]
7. Younis, Y.A.; Kifayat, K.; Merabti, M. An access control model for cloud computing. *J. Secur. Appl.* **2014**, *19*, 45–60. [CrossRef]
8. Cao, N.; Wang, C.; Li, M.; Ren, K.; Lou, W. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 222–233. [CrossRef]
9. Lacuesta, R.; Lloret, J.; Garcia, M.; Peñalver, L. Two secure and energy-saving spontaneous ad-hoc protocol for wireless mesh client networks. *J. Netw. Comput. Appl.* **2011**, *34*, 492–505. [CrossRef]
10. Ruj, S.; Stojmenovic, M.; Nayak, A. Privacy preserving access control with authentication for securing data in clouds. In Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Ottawa, ON, Canada, 13–16 May 2012; pp. 556–563. [CrossRef]
11. Chu, C.K.; Chow, S.S.; Tzeng, W.G.; Zhou, J.; Deng, R.H. Key-aggregate cryptosystem for scalable data sharing in cloud storage. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 468–477.
12. Do, J.M.; Song, Y.J.; Park, N. Attribute based proxy re-encryption for data confidentiality in cloud computing environments. In Proceedings of the 2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering (CNSI), Jeju Island, Korea, 23–25 May 2011; pp. 248–251.
13. Jiang, T.; Chen, X.; Wu, Q.; Ma, J.; Susilo, W.; Lou, W. Secure and Efficient Cloud Data Deduplication with Randomized Tag. *IEEE Trans. Inf. Forenics Secur.* **2017**, *12*, 532–543. [CrossRef]
14. Fabian, B.; Ermakova, T.; Junghanns, P. Collaborative and secure sharing of healthcare data in multi-clouds. *Inf. Syst.* **2015**, *48*, 132–150. [CrossRef]
15. Darwazeh, N.S.; Al-Qassas, R.S.; AlDosari, F. A secure cloud computing model based on data classification. *Procedia Comput. Sci.* **2015**, *52*, 1153–1158.
16. Somani, U.; Lakhani, K.; Mundra, M. Implementing digital signature with RSA encryption algorithm to enhance the Data Security of cloud in Cloud Computing. In Proceedings of the 2010 First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010), Solan, India, 28–30 October 2010; pp. 211–216.
17. Li, J.; Chen, X.; Huang, Q.; Wong, D.S. Digital provenance: Enabling secure data forensics in cloud computing. *Future Gener. Comput. Syst.* **2011**, *37*, 259–266. [CrossRef]
18. Xu, Z.; Kang, W.; Li, R.; Yow, K.; Xu, C.Z. Efficient multi-keyword ranked query on encrypted data in the cloud. In Proceedings of the 2012 IEEE 18th International Conference on Parallel and Distributed Systems, Singapore, 17–19 December 2012; pp. 244–251.
19. King, N.J.; Raja, V.T. Protecting the privacy and security of sensitive customer data in the cloud. *Comput. Law Secur. Rev.* **2012**, *28*, 308–319. [CrossRef]
20. Samanthula, B.K.; Elmehdwi, Y.; Howser, G.; Madria, S. A secure data sharing and query processing framework via federation of cloud computing. *Inf. Syst.* **2015**, *48*, 196–212. [CrossRef]
21. Kozak, S.; Novak, D.; Zezula, P. Secure metric-based index for similarity cloud. In Proceedings of the Workshop on Secure Data Management, Istanbul, Turkey, 27 August 2012; pp. 130–147.
22. Hong, H.; Sun, Z. An efficient and traceable KP-ABS scheme with untrusted attribute authority in cloud computing. *J. Cloud Comput.* **2016**, *5*, 1–8. [CrossRef]
23. Wang, H.; Wu, S.; Chen, M.; Wang, W. Security protection between users and the mobile media cloud. *IEEE Commun. Mag.* **2014**, *52*, 73–79. [CrossRef]
24. Li, J.; Huang, X.; Chen, X.; Xiang, Y. Securely outsourcing attribute-based encryption with checkability. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 2201–2210. [CrossRef]
25. Choi, C.; Choi, J.; Kim, P. Ontology-based access control model for security policy reasoning in cloud computing. *J. Supercomput.* **2014**, *67*, 711–722. [CrossRef]
26. Wei, L.; Zhu, H.; Cao, Z.; Dong, X.; Jia, W.; Chen, Y.; Vasilakos, A.V. Security and privacy for storage and computation in cloud computing. *Inf. Sci.* **2014**, *258*, 371–386. [CrossRef]
27. Li, M.; Yu, S.; Zheng, Y.; Ren, K.; Lou, W. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *24*, 131–143. [CrossRef]

28. Subashini, S.; Kavitha, K. A survey on security issues in service delivery models of cloud computing. *J. Netw. Comput. Appl.* **2011**, *34*, 1–11. [CrossRef]
29. Mousa, A. Data encryption performance based on Blowfish. In Proceedings of the 47th International Symposium ELMAR, 2005, Zadar, Croatia, 8–10 June 2015; pp. 131–134.
30. Huang, Q.; Yang, Y.; Shen, M. Secure and efficient data collaboration with hierarchical attribute-based encryption in cloud computing. *Future Gener. Comput. Syst.* **2017**, *72*, 239–249. [CrossRef]