Received January 27, 2020, accepted February 12, 2020, date of publication February 28, 2020, date of current version March 18, 2020.
Digital Object Identifier 10.1109/ACCESS.2020.2977264

An Improved Authentication Scheme for Remote Data Access and Sharing Over Cloud Storage in Cyber-Physical-Social-Systems

ZAHID GHAFFAR[1], SHAFIQ AHMED[1], KHALID MAHMOOD[1], SK HAFIZUL ISLAM[2], (Senior Member, IEEE), MOHAMMAD MEHEDI HASSAN[3], (Senior Member, IEEE), AND GIANCARLO FORTINO[4], (Senior Member, IEEE)

[1]Department of Computer Science, COMSATS University Islamabad at Sahiwal Campus, Sahiwal 57000, Pakistan
[2]Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, Kalyani 741235, India
[3]Research Chair of Smart Technologies, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[4]Department of Informatics, Modeling, Electronics, and Systems, University of Calabria, 87036 Rende, Italy

Corresponding author: Mohammad Mehedi Hassan (mmhassan@ksu.edu.sa)

This work was supported by the Deanship of Scientific Research at King Saud University through the Vice Deanship of Scientific Research Chairs. Chair of Smart Technologies.

ABSTRACT Cyber-physical-social systems (CPSSs) epitomize an evolving paradigm, including the social, physical, and cyber world. The vital goal of CPSSs is to offer personalized, high-quality, and proactive services for the end-users. An ingenious framework for reliable services is required for CPSSs to achieve this purpose. In this regard, the cloud storage environment of cloud computing (having a great connection with the physical, cyber, and social world) requires a reliable framework for secure communication between cloud and users. Cloud storage provides various services that need scalable, cost-effective, and secure facilities of data management. Public cloud storage bound its users to maintain strict security considerations that are offered by cloud service providers. On the other hand, an opportunity for users is offered by private cloud storage to construct a controlled and self-managed model of data security. This mobile model is responsible for managing the sharing and accessing of data privately. Despite that, it induces new challenges of data security. One censorious problem is to ensure the authenticated and secure model of data-storage for accessing the data under the controlled environment of data accessibility. To tackle this challenge, many protocols have been developed. The problem is that all these protocols are unable to fulfill the required security efficiency and are susceptible to various security attacks. Recently, Tiwari *et al.* presented an authentication scheme for data sharing and access with a biometric feature. They claimed that their scheme resists significant security attacks. However, in this article, we show that the claim of Tiwari *et al.* for developing a secure scheme is not valid, and their protocol is insecure against user and server impersonation attacks. Moreover, the protocol of Tiwari *et al.* do not provide user anonymity. Therefore, we present an enhanced, secure, and convenient scheme for data access. Besides, in order to add the flexible distribution of data that is controlled by data-owner, our protocol provides proxy re-encryption in which the cloud server utilizes the proxy re-encryption key. Then, the data-owner generates the credential token during decryption for controlling user's accessibility. The security analysis determines that our proposed protocol resists numerous security attacks. Furthermore, performance analysis determines that our protocol has practical computation, communication, and storage costs as compared to various related protocols. Consequently, our introduced protocol achieves not only the security goals but also has performance efficiency comparable to numerous relevant protocols of cloud storage.

INDEX TERMS Authentication, cloud storage, data sharing, data owner, impersonation attack.

I. INTRODUCTION

In In recent years, the hasty evolution of the Internet-of-Things (IoT) is specified as cyber-physical systems (CPSs), has increased the digital innovation and improved the

The associate editor coordinating the review of this manuscript and approving it for publication was Francesco Piccialli.

environment of human living. Cyber-physical-social systems (CPSSs) encompassing the physical, social, and cyber world, are escalating in every aspect of our daily lives. The provision of an efficient living environment by offering high quality personalized and proactive services to humans is one of the primary purposes of CPSSs [1]–[3]. The massive amount of data in our daily lives is constantly produced from cyber, physical, and social worlds. Data is the essential pivoting point of our research, which is flowing around these three worlds and maintaining patterns of all our daily life aspects. However, vast amounts of data acquired from CPSSs are usually complex, low-quality, noisy, and redundant, which causes unexceptional challenges for offering CPSSs services [1]. The comprehensive analysis of cloud computing based big data is essential. Moreover, secure communication for a cloud computing environment is also required to offer high quality and reliable services.

A new type of storage, referred to as cloud storage, has emerged with the evolution of the cloud computing paradigm. The adoption of cloud storage, particularly leveraging on the services of public or private cloud data storage, includes not only several benefits regarding reliability, flexibility, and scalability but also infers new challenges that are related to data privacy, protection, and security. Practically, public clouds provide several advantages such as no chain of risks towards infrastructure providers and no cost of the initial investment. However, efficient control on network, data, and security settings is lacked by the public that diminishes the interest of acquiring the services of the cloud. They also increase the challenges of data trust, security, and privacy. On the other hand, the private cloud reflects the quality of the service factors that includes reliability, security, and performance. Moreover, private cloud offers an opportunity to many organizations for the implementation of their own security acquiescence without depending on security measures of cloud providers. Whenever any sensitive data is involved, the services of a private cloud storage are used. The model of private cloud storage has following concerns that are related to operations of data management:

(1). Insider attack and security perimeter: An organization develops its own secure and private network by defining criteria of perimeter security that permits only authorized users to access several resources that include network bandwidth, network computation, and network storage through a configured network. Moreover, the toughest problem in cloud storage is to establish protection against the insider attack [4]. Therefore, cloud storage systems need an authentication model of data-access. This model allows only authorized users to register their permission.

(2). Lack of the verification model accessing data: In cloud storage, only user-name and password are inefficient to secure communication of data. Therefore, the cloud storage environment should accommodate a flexible, multi-factor, and maintainable system of data-access.

(3). Lack of configuration for sharing data: A system of data-sharing is dependent on a one-to-one or all-or-nothing model of sharing that is insufficient for offering the sharing of data under data delegator's control. Therefore, the system of cloud storage requires an efficient system of data-access controlled by data-owner ($\mathcal{DO}$) that reflects several security perimeters that includes verifiability, correctness, confidentiality, and non-transferability of data.

To tackle the above-discussed concerns of security, a $\mathcal{DO}$-controlled and biometric-based system of data-access is needed for the development of a private and secure storage model. For sharing the data, the $\mathcal{DO}$ should take part in that process, to manage the convenience of users and only authorized access should be allowed in the model of cloud storage. However, the methods of data sharing (i.e., proxy encryption [5], [6] and Attribute Base Encryption (ABE) [7], [8]), which are used in the public cloud storages, are not convenient for use in storage model of private cloud. The reason is that the $\mathcal{DO}$ is not aware of the specific user's identity user who needs to acquire data. The $\mathcal{DO}$ also provides a secure mechanism of group data-sharing.

The FIGURE 1 illustrates the architecture of data access and sharing model for the environment of cloud storage, which includes three parties: the data-owner ($\mathcal{DO}$), mobile user $\mathcal{MU}_a$ and the cloud server $\mathcal{CS}$.

(1). $\mathcal{DO}$ performs the encryption of data before it is stored in cloud storage. During the request of data-access service, it controls $\mathcal{MU}_a$ accessibility for accessing encrypted data. Before storing data in the cloud, $\mathcal{DO}$ invokes the system of data-access for establishing a mutual authentication mechanism with $\mathcal{CS}$ for authentication purposes.

(2). $\mathcal{CS}$ implements various services of cloud storage that include data-storage, data-access, and data-sharing. The service of data-access offers biometric authentication for enabling secure and authorized communication with the server of cloud storage. Moreover, this service provides a mutual authentication facility before downloading or uploading data on cloud storage. Service of data-storage helps $\mathcal{DO}$ for uploading encrypted data with a key of proxy re-encryption to offer convenient control of data access. The services of data-sharing offers accredited access of data to $\mathcal{MU}_a$ under $\mathcal{DO}$'s control by producing credential token.

(3). Whenever, $\mathcal{MU}_a$ requests to access particular data, then the authentication of $\mathcal{MU}_a$ is done by a data-access system that allows $\mathcal{MU}_a$ to invocate the system of data-sharing. The proxy encryption key, used by the data-sharing system, changes the encrypted data in such a way that it can be easily decrypted by $\mathcal{MU}_a$ using credential token. A credential token is engendered by $\mathcal{DO}$ for permitting $\mathcal{MU}_a$ that performs decryption of re-encrypted data.

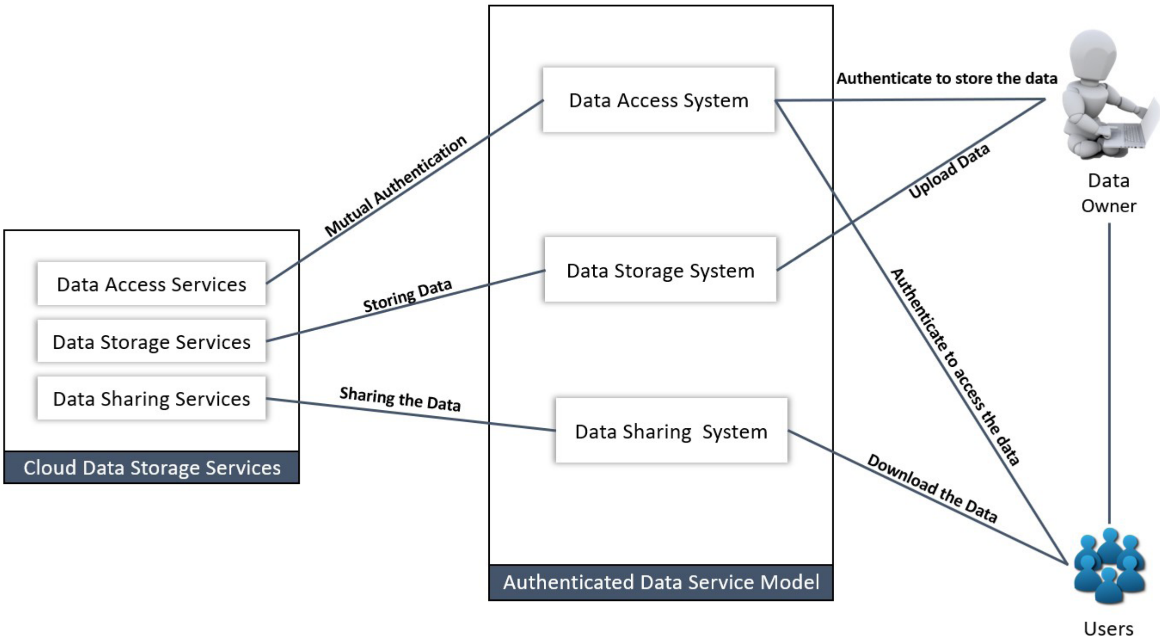

FIGURE 1. **Controlled data access and sharing system.**

A. CONTRIBUTIONS

We present a model of data-access that allows only a legal $\mathcal{MU}_a$ to access the data by log-in to the cloud server with a multi-factor authentication mechanism using a password and biometric as input. Moreover, we have defined a $\mathcal{DO}$-controlled access system that uses the proxy re-encryption mechanism. Further, $\mathcal{DO}$ permits controlled permission for $\mathcal{MU}_a$ to access data by computing credential token. We described our main contributions below:

(1). Our proposed protocol involves a biometric-based system that provides an authenticated system of data-access and uniquely finds the identity on a $\mathcal{MU}_a$. Mainly, before downloading or uploading of data, our system of data-access establishes mutual authentication by setting up a session key between the cloud server and $\mathcal{MU}_a$ for achieving secure communication.

(2). In order to provide a secure and flexible system of data storage, the symmetric key is derived by $\mathcal{DO}$ using the system of hash-keying on the input of the file. It assimilates the decipher key that is encrypted into meta-data for ensuring data confidentiality and de-duplication. Our proposed protocol derives a key to proxy re-encryption to provide access control over encrypted data. On the other side, our proposed protocol offers controlled sharing of data with a legal $\mathcal{MU}_a$ by generating a credential token of the $\mathcal{MU}_a$.

(3). Our protocol uses the idea of proxy re-encryption for the flexible sharing of data with $\mathcal{MU}_a$. It also uses convergent encryption [9] for defining and managing the accessibility of legal $\mathcal{MU}_a$.

We have arranged this article in the following manner. Related work about the cloud environment is discussed in Section II. Section III describes the complexity assumptions and definitions, whereas, security requirements for the cloud environment are briefly described in Section IV. The protocol of Tiwari *et al.* is reviewed in Section V. The cryptanalysis of Tiwari *et al.'s* protocol [10] is explained in Section VI. The proposed protocol is described in Section VII. The security analysis of the proposed protocol, which includes the informal and formal security analysis, is specified in Section VIII. Moreover, Section IX shows the theoretical analysis, and Section X concludes this article.

II. RELATED WORK

The numerous collection of technologies configures cloud storage, and the cloud server manages it. So, the security of outsourced data relies on the flexible and secure data services in cloud storage that facilitates data security, privacy, and authentication to a user for different operations of data. The data distribution, data storage, and data access are such components of data service which are required for cloud storage to give secure solutions such as auditing, integrity, and flexible sharing of data [11], [12].

The existing solutions for cloud storage are concentrated on offering an efficient access control method to outsourced data by introducing a key management system. For flexible data access, this system enforces a self-reliant authorization with legitimate users [13], [14]. The job of data owner is different from the providers of cloud storage due to the outsourcing of data service management, and the users do not interact with the data owner for offering authorized data access. So, numerous protocols, i.e., attribute-based encryption [7], [8], [15], [16], proxy-based encryption [5], [17]–[19] and user-based access control protocols [20], [21] have been utilized to offer self-reliant, flexible, and secure outsourced encrypted data. Kamara and Lauter [20] develop a user-based access-control protocol for cloud storage with the help of symmetric cryptography. In this approach, the data owner predetermines every user, and the data owner for data decryption issues the relevant key. However, this protocol [20] is not able to support the configuration of a user dynamically because the symmetric key is shared with every user who sends the request at that time.

Zhao *et al.* [21] proposed a protocol that shares data by utilizing elliptic curve encryption to achieve a flexible configuration of data access on an unreliable storage server. In this type of encryption, multiple keys are used to encrypt the data and help to decipher the encrypted data with a single key. However, the secret key of the data owner is required to share with cloud storage for data re-encryption. Thus, this activity is not efficient in cloud storage for data access, as it affects privacy. Furthermore, these protocols [20], [21] has a high cost of communication and computation for revocation because the data owner fetches the complete data set. Furthermore, when the data owner needs to redistribute that data to the cloud server then the data needs to be re-encrypted. An architecture for data storage is developed by Kumbhare *et al.* [22] to store the scientific data in the cloud, and they associate this scheme with a controlled data access scheme to share the data with numerous users. However, random file sharing is not supported by this scheme [22].

The protocols based on ciphertext policy attribute-based encryption [23]–[25] are such kinds of cryptographic approaches in which ciphertext is combined with controlled access method and attributes are combined with the private key of the user. If the association of attributes with the secret key of users satisfies the access policy of ciphertext, the ciphertext is decrypted by the user. In this way, the data owner can control the sharing of information by introducing the distribution model without knowing the user's identity. So, the application of ciphertext policy attribute-based encryption is very appropriate in public cloud for data distribution. Still, it cannot represent a user-based access mechanism where the data owner usually permits every user to acquire data from a private cloud. In proxy re-encryption, the data owner has no control over the users who require to get the data. So, the data owner is not able to define the flexible system of data access. Therefore, a variant approach for proxy re-encryption is required, where along with the re-encryption key, the data owner determines the credential token for the request of the user to permit the decryption of data. Sultan *et al.* [26] introduced a protocol for flexible data distribution with reliable access to data. However, authenticated access to data is not offered by this protocol that can facilitate validation to the authentic users before performing different operations of data such as storage, share and update. Recently, Tiwari *et al.* [10] developed a protocol for a cloud storage environment. We determined that the protocol [10] does not offer user anonymity. Moreover, Tiwari *et al.'s* protocol is susceptible to mobile and cloud server impersonation attacks.

Therefore, the flaws and vulnerabilities of the above-discussed schemes incite us to develop an improved protocol having the ability to resist various attacks and remove all these flaws specially present in Tiwari *et al.'s* scheme.

III. PRELIMINARIES

Basic cryptographic primitives including elliptic curve cryptography, hash function, bilinear map, and fuzzy extractor are described below in detail.

A. HASH FUNCTION

A function of hash $h : \{0, 1\}^* \rightarrow \mathcal{Z}_p^*$ develops an output of fixed size $z = h(st)$ when it takes a random binary string st as input. The achieved output is hash code or hash value. A minor change in the value of the input string can make a major difference in the output. For a protected hash function, subsequent parameters must be achieved.

- **(1).** For any given input st, the computation of $h(st)$ is usually easy.
- **(2).** **One-way property:** From a given value of hash $h(st)$ and a function $h(\cdot)$, it is hard to obtain the input st. This property is called ***preimage resistance***.
- **(3).** **Resistance property of weak-collision:** For any given input st_1, obtaining any other input st_2, with $st_1 \neq st_2$, such that $h(st_1) = h(st_2)$. It is hard to determine. This property is called ***second preimage resistance***.
- **(4).** **Resistance property of strong-collision:** Deriving a pair of inputs (st_1, st_2), with $st_1 \neq st_2$, such that $h(st_1) = h(st_2)$ is also hard to calculate.

B. BILINEAR MAPS

Assume that $E(G_k)$ is a set of elliptic curve points for the elliptic curve $y^2 \bmod p \equiv (x^3+ax+b) \bmod p$ over the field F, where $a, b \in \mathcal{Z}_p^*$ and $(4a^3+27b^2) \bmod p \neq 0$. The set $E(G_k)$ together with the "*point of infinity*" $\mathcal{O}$ defines an additive cyclic group A_1 of prime order p. Let A_2 a multiplicative group of the same order p. The bilinear map $e : A_1 \times A_1 \rightarrow A_2$ must satisfy the following properties:

1) **Bilinearity:** $\forall\, X, Y, Z \in A_1$ and $c, d \in Z_q$, $e(Y, X + Z) = e(X, Y) \cdot e(Z, Y)$. Moreover, $e(X, dY) = e(X, dQ)^c = e(cX, Y)^d = e(X, Y)^{cd}$.
2) **Non-degeneracy:** If X is a generator of A_1, then $(X, X) \neq 1$, where 1 is the generator of A_2.
3) **Computability:** It is efficient to compute $e(X, Y)$ $\forall$ $X, Y \in A_1$.

C. DEFINITION OF ECC (ELLIPTIC CURVE CRYPTOGRAPHY)

The equation of elliptic curve is specified in the form: $E_p(a, b)$: $y^2 \bmod p \equiv (x^3 + ax + b) \bmod p$. See Section B. Bilinear Maps over a field of prime finite F_p, where $a, b \in F_p$ and $4a^3 + 27b^2 \neq 0 \bmod p$ See Section B. Bilinear Maps. In general, the ECC security relied on following problems which are infeasible for computation.

1) DEFINITION OF ECDLP (ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM)

Consider F be the subgroup of $E(G_k)$, which is being produced by point P of prime order p. Given a point P, which is a generator of an additive cyclic group A of order p, and $R = aP \in F$ the public element, problem of computing $a \in \mathcal{Z}_p^*$ is called ECDLP [27].

2) DEFINITION OF ECCDH (ELLIPTIC CURVE COMPUTATIONAL DIFFIE HELLMAN PROBLEM)

Suppose a cyclic group A of order p, with a generator P and randomly chosen numbers $c, d \in \mathcal{Z}_p^*$. It is computationally infeasible to compute cdP for input pair of (P, cP, dP) is given. We can state as, $\mathcal{U}_A$ has the benefit ϵ in polynomial time for computing CDH in $\langle A, p, P\rangle$ if: $\mathbf{Pr}[\mathcal{U}_A(P, cP, dP) = cdP] \leq \epsilon$, whereas the probability is find out over the bits used by an $\mathcal{A}_{adv}$ and randomly chosen $c, d \in \mathcal{Z}_p^*$ [28].

D. ADVERSARIAL MODEL

In this section, the basic assumptions about the attacker are described according to which he breaches the system or can launch attacks. The assumptions about adversary are as follows:

1) An attacker can be a server, user, or a gateway. The user, which is already registered at the system, can play the role of attacker.
2) The communication which is exchanged on the public communication channel can be intercepted by the attacker.
3) An attacker can delete, transform, or replay the intercepted information.
4) An attacker can fetch the information saved in the smart card by doing power analysis.
5) An attacker cannot change, shift, replay, or obstruct the information exchanged on the private communication path.

E. FUZZY EXTRACTOR

The biometric information of user is converted into arbitrary, secrete and reproducible string of size n, it is also used in cryptographic functions which are used to authenticate user with error tolerance limit $\mathcal{T}$. Suppose $N = \{0, 1\}^d$, d is a biometric point in dimensional metric space with distance function $D : N \times N \rightarrow \mathcal{Z}^+$, with the help of given metric, it calculates the distance between two biometric points. For authenticating users, two main fuzzy extractor procedures **Gen(·)** and **Rep(·)** are used which are defined below:

1) **Gen(·):** It is a probabilistic generation method that takes input ($\mathcal{B}_u \in N$) biometric information of user and generates $\alpha_u \in \{0, 1\}^n$ a secrete string along with τ_u an associative string, i.e., $Gen(\mathcal{B}_u) = \{\alpha_u, \tau_u\}$
2) **Rep(·):** It is deterministic reproduction method that takes a noisy input (biometric) $\mathcal{B}'_u$, a public string τ_u and reproduce α_u a secrete string, i.e., $Rep(\mathcal{B}'_u, \tau_u) = \alpha_u$, if $D(\mathcal{B}_u, \mathcal{B}'_u) \leq \mathcal{T}$

IV. SECURITY REQUIREMENTS

In this section, we state the security requirements required for the authenticated controlled data access and sharing schemes for the cloud.

A. SECURITY REQUIREMENTS IN DATA ACCESS SYSTEM

The data access system communicates with data access service for authenticating the user before giving access to the user or $\mathcal{DO}$ for data sharing services and data storage services. The following properties must be satisfied by the data access system for authenticating a user.

1) **Non-Repudiation:** It ensures that participants (i.e., $\mathcal{MU}_a$, $\mathcal{CS}$ and $\mathcal{DO}$) can not deny their authenticity after executing any operation.
2) **Strong Authentication:** Strong user identification mechanism should be provided by the protocol for proving the user's authenticity before giving access to data services.
3) **User-friendliness and Scalable:** User-friendliness refers to that user can change the password. User identification is based on a multi-factor authentication mechanism, whereas; data access system should support authenticated dynamic user configuration.
4) **Accountability:** Accountability guarantees that the data operations performed by the participants can be identified without compromising the privacy of the participants (i.e, $\mathcal{MU}_a$, $\mathcal{CS}$ and $\mathcal{DO}$)
5) **Security against Various Attacks:** The system should resist major security attacks, particularly impersonation, password guessing, and replay attack.

B. SECURITY REQUIREMENTS IN THE DATA STORAGE AND SHARING SYSTEM

The on and off-premises storage system is provided by the cloud provider to manage and store the data. The cloud provider is assumed to be curious but honest, and the operation will be performed as per the steps described in the protocol. However, by learning the encrypted data content and by colluding with other participants, data privacy can be compromised by the cloud provider. Thus, data sharing and storage system must fulfill following requirements:

1) **Data Confidentiality:** Cloud data should be in encrypted form to secure it from unauthorized access.
2) **Deduplication:** For reducing the storage overhead, the storage system should discard duplicate copies of data. It ensures that only one instance of the data is maintained in the memory.
3) **Flexible Data Access Control:** For a different group of users, data owner $\mathcal{DO}$ should specify the flexible data access policies according to their roles.
4) **Data Integrity:** The data storage system should prevent any unintended changes in the data from an unauthorized user. The identification of changes and originality of the data can be verified by the verifier or $\mathcal{DO}$.
5) **Verifiability:** The data storage system should verify the identity of $\mathcal{DO}$ before storing data in the cloud. The permission should be given by the $\mathcal{DO}$ to the user to decrypt the data during the data sharing process.
6) **Collusion between Entities:** With the data owner's permission, the only legal user, should be able to access

TABLE 1. Common used notations.

Notation	Description
$\mathcal{MU}_a$	a^{th} user of the system
$\mathcal{CS}$	Cloud server of the system
ID_a	Identity of $\mathcal{MU}_a$
PW_a	Password of $\mathcal{MU}_a$
BIO_a	Biometric of $\mathcal{MU}_a$
ID_{cs}	Identity of $\mathcal{CS}$
T_{S_k}	The session token
S_k	Session key between $\mathcal{CS}$ and $\mathcal{MU}_a$
sk_{cs},pk_{cs}	Private and public key of $\mathcal{CS}$
D_f	Data file
$D_f^{\prime}$	Encrypted data file
$\langle A_1, A_2\rangle$	Proxy re-encryption token
$\langle A_3, A_4\rangle$	Credentials token
F_{em}	First layer encrypted message
S_{em}	Second layer encrypted message
S_{key}	Symmetric encryption key
$\mathcal{MD}_a$	Mobile device of $\mathcal{MU}_a$
P	Elliptic curve point
$\mathcal{U}_A$	Adversary
$h(\cdot)$	One-way hash function
$\oplus$	XOR operator
$\|$	Concatenation operator
$\Longrightarrow$	Secure channel
$\longrightarrow$	Public channel

the data from the data sharing system. It states that colluding entities without data owner's permission, can not access any data.

The notations or symbols used in this manuscript are described in TABLE 1.

V. REVIEW OF TIWARI *et al.'s* SCHEME

In this section, the description of Tiwari *et al.* [10] protocol is illustrated.

A. DATA ACCESS PHASE

The data access service is a profound element of cloud storage bound to empower a solid confirmation to the accredited users who stock, share and amend data by accessing the data service. The modern-day data access systems are known to have low security and contain a potential threat of being easily accessed by hackers. Generally, it is because they are based on simple passwords and private data tokening. Many potential risks can be characterized as phishing attacks, replay attacks, password guessing, and denial of service. The biometric authentication model is considered a reliable tool that provides security features by uniquely identifying the user's identity. Therefore, the suggested data access system is a user authentication system for secure access to data service in the cloud with the following steps:

1) REGISTRATION PHASE

The user registers himself with $\mathcal{CS}$, in this phase. FIGURE 2 illustrates the complete description of registration phase. The following steps are performed by $\mathcal{CS}$ and user, to determine the information of registration.

Step 1: The user $\mathcal{MU}_a$ selects his identity ID_a, password PW_a, and biometric BIO_a. The input of biometric information BIO_a is taken by fuzzy extractor embedded in the mobile device ($\mathcal{MD}_a$). The $\mathcal{MD}_a$ engenders an arbitrary number r_{a1} and determines $PWP_a = h_1(ID_a\|PW_a\|E_{a1}\|r_{a1})$. Then the ID_a, PWP_a is sent to $\mathcal{CS}$ by $\mathcal{MD}_a$ for registering the $\mathcal{MU}_a$.

Step 2: The $\mathcal{CS}$ calculates $O_a = h_1(ID_a\|r_{cs1})p$, $P_a = PWP_a \oplus h_1(ID_a \oplus s_{cs1})$, $Q_a = h_1(ID_a\|PWP_a\| h_1(ID_a \oplus s_{cs1}))$ and $R_a = h_1(ID_a \oplus PWP_a) \oplus O_a$ on receiving request by $\mathcal{MU}_a$, where r_{cs1} and s_{cs1} are random numbers. The $\mathcal{CS}$ sends $\{P_a, Q_a, R_a\}$ to $\mathcal{MU}_a$.

Step 3: The registration parameters (P_a, Q_a, R_a, r_{a1}, $Gen(\cdot)$, $Rep(\cdot)$) are stored into $\mathcal{MD}_a$ by $\mathcal{MU}_a$ on receiving $\{P_a, Q_a, R_a\}$.

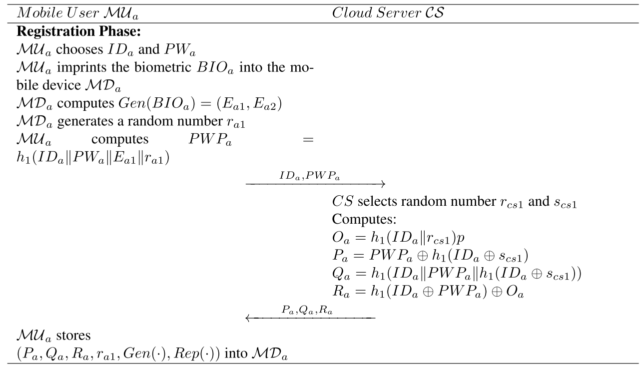

FIGURE 2. Registration phase of Tiwari *et al.'s* scheme.

2) LOGIN AND AUTHENTICATION PHASE

In this phase, $\mathcal{MU}_a$ makes a data access request to the $\mathcal{CS}$ by entering login information using a verified device MD_a. MD_a checks and verifies the identity of $\mathcal{MU}_a$ before sending a request to $\mathcal{CS}$. Then the login request of $\mathcal{MU}_a$ is processed by $\mathcal{CS}$ and makes a mutual connection between MD_a and $\mathcal{CS}$ only if $\mathcal{MU}_a$ is a legal user. The data access operations such as storing, sharing, and modification of data in the cloud are performed when $\mathcal{CS}$ verifies and establishes the session key by $\mathcal{MU}_a$'s authorization. The detailed description of login and authentication phase is shown in FIGURE 3. To set up a session key, $\mathcal{MU}_a$ and $\mathcal{CS}$ need to verify each other as follows:

Step 1: On log-in screen, $\mathcal{MU}_a$ enters ID_a and PW_a, and punch the biometric information $BIO_a^{\prime}$ on fuzzy extractor embded into MD_a. MD_a then calculates $Rep(BIO_a^{\prime}, E_{a2}) = (E_{a1})$, $PWP_a^{\prime} = h_1(ID_a\|PW_a\|E_{a1}\|r_{a1})$, $h_1(ID_a \oplus s_{cs1})^{\prime} = P_a \oplus PWP_a$ and $Q_a^{\prime} = h_1(ID_a\|PWP_a\|h_1(ID_a \oplus s_{cs1})^{\prime})$ for $\mathcal{MU}_a$'s verification. For verifying the registration of $\mathcal{MU}_a$ with $\mathcal{CS}$, the MD_a checks $Q_a \stackrel{?}{=} Q_a$. MD_a calculates $O_a = h_1(ID_a\|PWP_a) \oplus R_a$, if the condition is true. For computing $X_a = r_{a2}P, X_a^{\prime} = r_{a2}O_a$ and $Y_a = h_1(ID_a\|X_a\|X_a^{\prime}\|T_a)$, MD_a engenders an arbitrary nonce $r_{a2} \in Z_q^*$ and obtains the current time stamp T_a. MD_a sends the message $\{ID_a, X_a, Y_a, T_a\}$ to $\mathcal{CS}$.

Mobile User $\mathcal{MU}_a$	Cloud Server $\mathcal{CS}$
Login and Authentication Phase:	
$\mathcal{MU}_a$ selects ID_a, PW_a and imprints BIO'_a on $\mathcal{MD}_a$	
$\mathcal{MD}_a$ computes:	
$Rep(BIO'_a, E_{a2}) = (E_{a1})$	
$\mathcal{MD}_a$ computes:	
$PWP'_a = h_1(ID_a \| PW_a \| E_{a1} \| r_{a1})$	
$h_1(ID_a \oplus s_{cs1})' = P_a \oplus PWP_a$	
$Q'_a = h_1(ID_a \| PWP_a \| h_1(ID_a \oplus s_{cs1})')$	
Checks whether the user registered with $\mathcal{CS}$	
if $Q_a \stackrel{?}{=} Q_a$ is true, then calculates	
$O_a = h_1(ID_a \| PWP_a) \oplus R_a$	
Obtains the current timestamp T_a	
Selects the random number $r_{a2} \epsilon Z_q^*$	
Calculates:	
$X_a = r_{a2}P, X'_a = r_{a2}O_a$	
$Y_a = h_1(ID_a \| X_a \| X'_a \| T_a)$	
$\xrightarrow{ID_a, X_a, Y_a, T_a}$	
	Verifies $T_a - T_{cs} \leq \Delta T$
	$X'_a = h(ID_a \| r_{cs1})X_a$
	$Y_a \stackrel{?}{=} (Y'_a = h_1(ID_a \| X_a \| X'_a \| T_a))$
	Selects random number $r_{cs2} \in Z_q^*$
	$X_{cs} = r_{cs2}P, T_{S_k} = r_{cs2}X_a$
	$S_k = h_1(T_{S_k})$
	$Y_{cs} = h_1(ID_a \| ID_{cs} \| X_a \| X'_a \| T_a \| T_{cs} \| T_{S_k})$
$\xleftarrow{ID_a, X_{cs}, Y_{cs}, T_{cs}}$	
Verifies whether $T_{cs} - T_a \leq \Delta T$	
Calculates: $T_{S_k} = r_{a1}X_{cs}$	
Verifies:	
$Y_{cs} \stackrel{?}{=} (Y'_{cs} = h_1(ID_a \| ID_{cs} \| X_a \| X'_a \| T_a \| T_{cs} \| T_{S_k})$	
Establishes the session key $S_k = h_1(T_{S_k})$ with $\mathcal{CS}$	

FIGURE 3. Login and authentication phase of Tiwari *et al.'s* scheme.

Step 2: $\mathcal{CS}$ verifies the transmission delay $T_a - T_{cs} \leq \Delta T$, after receiving the message $\{ID_a, X_a, Y_a, T_a\}$. After that, the $\mathcal{CS}$ determines $X'_a = h(ID_a \| r_{cs1})X_a$. Then, $\mathcal{CS}$ calculates $Y_a \stackrel{?}{=} (Y'_a = h_1(ID_a \| X_a \| X^*_a \| T_a))$ and checks whether $Y_a \stackrel{?}{=} Y'_a$. The session is terminated if the said condition becomes wrong, otherwise the $\mathcal{CS}$ verifies $\mathcal{MU}_a$. $\mathcal{CS}$ engenders a random number $r_{cs2} \epsilon Z_q$ and calculates $X_{cs} = r_{cs2}P$, $T_{S_k} = r_{cs2}X_a$, the session token $S_k = h_1(T_{S_k})$ and $Y_{cs} = h_1(ID_a \| ID_{cs} \| X_a \| X'_a \| T_a \| T_{cs} \| T_{S_k})$. Then, $\mathcal{CS}$ sends the message $\{ID_a, X_{cs}, Y_{cs}, T_{cs}\}$ to $\mathcal{MU}_a$.

Step 3: $\mathcal{MU}_a$ calculates the $T_{cs} - T_a \leq \Delta T$, when receives the message to verify the transmission delay, where T_a is the current time stamp of $\mathcal{MU}_a$. A session token $T_{S_k} = r_{a1}X_{cs}$ is calculated if the said condition becomes true. To agree on a common session key S_k, $\mathcal{MU}_a$ checks $Y_{cs} \stackrel{?}{=} (Y'_{cs} = h_1(ID_a \| ID_{cs} \| X_a \| X'_a \| T_a \| T_{cs} \| T_{S_k}))$. The session is ended by $\mathcal{MU}_a$ if the said condition becomes false, otherwise, $\mathcal{MU}_a$ confirms the verification of $\mathcal{CS}$. $\mathcal{MU}_a$ builds up mutual authentication with $\mathcal{CS}$ after calculating the session key $S_k = h_1(T_{S_k})$.

3) PASSWORD CHANGE PHASE

This stage offers the user to freely selects any password and inform the $\mathcal{CS}$ with newly selected password without involving the $\mathcal{CS}$. $\mathcal{MU}_a$ can log-in to the system after entering ID_a, PW_a and punch BIO_a on the fuzzy extractor of $\mathcal{MD}_a$. $\mathcal{MD}_a$ calculates $PWP_a = h_1(ID_a \| PW_a \| E_{a1} \| r_{a1})$, $h_1(ID_a \oplus s_{cs1}) = P_a \oplus PWP_a$ and $Q'_a = h_1(ID_a \| PWP_a \| h_1(ID_a \oplus s_{cs1}))$. Moreover, $\mathcal{MD}_a$ verifies the registration of $\mathcal{MU}_a$ with $\mathcal{CS}$ by computing $Q'_a \stackrel{?}{=} Q_a$. $\mathcal{MD}_a$ cancels the password change request, if the said condition is false. Otherwise it allows $\mathcal{MU}_a$ to set new password PW_a^{new} and requests the system to update it. In order to successfully inform the system with new password, $\mathcal{MD}_a$ calculates $PWP_a^{new} = h_1(ID_a \| PW_a^{new} \| E_{a1} \| r_{a1})$, $P_a^{new} = h_1(PWP_a \oplus h_1(ID_a \oplus s_{cs1}))$, $Q_a^{new} = h_1(ID_a \| PWP_a^{new} \| h_1(ID_a \oplus s_{cs1}))$ and $O_a^{new} = h_1(ID_a \| PWP_a^{new}) \oplus R_a$ and changes the value of P_a, Q_a and O_a with P_a^{new}, Q_a^{new} and O_a^{new}.

VI. CRYPTANALYSIS OF TIWARI *et al.'s* SCHEME

In this section, we have presented the cryptanalysis of Tiwari *et al.'s* scheme. The scheme of Tiwari *et al.* is vulnerable to cloud server and mobile user impersonation attacks. Moreover, it does not offer user anonymity.

A. NO PROVISION FOR USER ANONYMITY

As ID_a of $\mathcal{MU}_a$ is sent in plaintext to the cloud server $\mathcal{CS}$ through a public channel. Therefore, the scheme of Tiwari *et al.* does not offer user anonymity because an adversary can easily determine the user's identity by just intercepting the login request message $\langle ID_a, X_a, Y_a, T_a \rangle$. This leakage of identity can facilitate the adversary to trace the current location or login history of the device.

B. CLOUD SERVER IMPERSONATION ATTACK

In the registration phase, the value of r_{cs1} is generated randomly for the calculation of O_a, which will be different for each user. Furthermore, in login and authentication phase, $\mathcal{CS}$ uses the same information r_{cs1} in order to determine true value of $X_a^* = h(ID_a \| r_{cs1})X_a$. As the value of $r_{cs,1}$ is different in the registration phase for each user. Therefore, if the server uses it later in the login and authentication phase then it must be stored in some repository. Assume that an adversary $\mathcal{U}_A$ obtains r_{cs1} from the repository via stolen verifier attack, then he/she can impersonate $\mathcal{CS}$ easily on the behalf of $\mathcal{CS}$ by following these steps as shown below:

Step 1: Whenever $\mathcal{MU}_a$ sends a login request message $\langle ID_a, X_a, Y_a, T_a \rangle$ to $\mathcal{CS}$, $\mathcal{U}_A$ intercepts the message and calculates the following values after verifying the freshness of timestamps:

$$X_a^{*'} = h_1(ID_a \| r_{cs1})X_a \tag{1}$$

Step 2: Afterwards, $\mathcal{U}_A$ chooses an arbitrary number $r_{cs2}^* \epsilon Z_q^*$ and calculates the following:

$$X_{cs}^* = r_{cs2}^*P, T_{S_k}^* = r_{cs2}^*X_a \tag{2}$$
$$S_k = h_1(T_{S_k}^*) \tag{3}$$

Step 3: After the above calculation, $\mathcal{U}_A$ calculates the signature of $\mathcal{CS}$ as follows:

$$Y_{cs}^* = h_1(ID_a \| ID_{cs} \| X_a \| X_a^{*'} \| T_a \| T_{cs}^* \| T_{S_k}) \tag{4}$$

Step 4: Finally, $\mathcal{U}_A$ sends a challenge message $\langle ID_a, X_{cs}^*, Y_{cs}^*, T_{cs}^* \rangle$ back to $\mathcal{MU}_a$.

Step 5: Upon receiving $\langle ID_a, X_{cs}^*, Y_{cs}^*, T_{cs}^* \rangle$, $\mathcal{MU}_a$ verifies the freshness of timestamps and calculates the

Mobile User $\mathcal{MU}_a$	*Adversary* $\mathcal{U}_A$
Login and Authentication Phase:	
$\xrightarrow{ID_a, X_a, Y_a, T_a}$	
	Verifies $T_a - T_{cs} \leq \Delta T$ $X_a^{*'} = h_1(ID_a \| r_{cs1})X_a$ $Y_a \stackrel{?}{=} (Y_a' = h_1(ID_a \| X_a \| X_a' \| T_a))$ Selects random number $r_{cs2}^* \epsilon Z_q^*$ $X_{cs}^* = r_{cs2}^*P, T_{S_k}^* = r_{cs2}^*X_a$ $S_k = h_1(T_{S_k}^*)$ $Y_{cs}^* = h_1(ID_a \| ID_{cs} \| X_a \| X_a^{*'} \| T_a \| T_{cs}^* \| T_{S_k})$
$\xleftarrow{ID_a, X_{cs}^*, Y_{cs}^*, T_{cs}^*}$	
Verifies whether $T_{cs}^* - T_a \leq \Delta T$ Calculates: $T_{S_k} = r_{a1}X_{cs}^*$ Verifies: $Y^*{}_{cs} \stackrel{?}{=} (Y_{cs}' = h_1(ID_a \| ID_{cs} \| X_a \| X_a' \| T_a \| T_{cs}^* \| T_{S_k})$ Establishes the session key $S_k = h_1(T_{S_k})$ with $\mathcal{CS}$	

FIGURE 4. Cloud server impersonation attack.

following:

$$T_{S_k} = r_{a1}X_{cs}^*, \tag{5}$$
$$Y_{cs}^* \stackrel{?}{=} (Y_{cs}' = h_1(ID_a \| ID_{cs} \| X_a \| X_a' \| T_a \| T_{cs}^* \| T_{S_k})). \tag{6}$$
$$S_K = h_1(T_{S_k}) \tag{7}$$

Hence, $\mathcal{U}_A$ can successfully impersonate $\mathcal{CS}$. Therefore, this scheme is prone to the server impersonation attack. The detailed demonstration of cloud server impersonation attack is also shown in FIGURE 4.

C. MOBILE USER IMPERSONATION ATTACK

An adversary $\mathcal{U}_A$ can easily get the identity ID_a of the legal mobile user $\mathcal{MU}_a$ as shown in section VI-A. Furthermore, $\mathcal{U}_A$ can also extract r_{cs1} of that particular mobile user $\mathcal{MU}_a$ corresponding to mobile user's ID_a as discussed in section VI-B. After having ID_a and r_{cs1} of a legitimate mobile user $\mathcal{MU}_a$, $\mathcal{U}_A$ can easily impersonate $\mathcal{MU}_a$. In order to get services as a legitimate user, $\mathcal{U}_A$ has to perform following these steps as follows:

Step 1: $\mathcal{U}_A$ calculates $O_a = h_1(ID_a \| r_{cs1})P$.

Step 2: Afterward, $\mathcal{U}_A$ generates a random number $r_{a2}^* \in Z_q^*$ and calculates the followings:

$$X_a^* = r_{a2}^*P, X_a^{*'} = r_{a2}^*O_a \tag{8}$$
$$Y_a^* = h_1(ID_a \| X_a^* \| X_a^{*'} \| T_a^*) \tag{9}$$

Step 3: After the above calculations, $\mathcal{U}_A$ sends request message $\langle ID_a, X_a^*, Y_a^*, T_a^* \rangle$ to $\mathcal{CS}$.

Step 4: Upon receiving $\langle ID_a, X_a^*, Y_a^*, T_a^* \rangle$, $\mathcal{CS}$ firstly verifies the freshness of timestamp T_a^* and calculates the following:

$$\acute{X}_a = h_1(ID_a \| r_{cs1})X_a^*, \tag{10}$$
$$Y_a^* \stackrel{?}{=} (Y_a = h_1(ID_a \| X_a^* \| \acute{X}_a \| T_a^*)) \tag{11}$$

Step 5: If above calculation holds true, then $\mathcal{CS}$ selects a random number $r_{cs2} \in Z_q^*$ and continues further calculation as follows:

$$X_{cs} = r_{cs2}P \tag{12}$$
$$T_{S_k} = r_{cs2}X_a^* \tag{13}$$
$$S_k = h_1(T_{S_k}) \tag{14}$$

Adversary $\mathcal{U}_A$		*Cloud Server* $\mathcal{CS}$
Login and Authentication Phase:		
Obtains the current timestamp T_a Selects the random number $r_{a2} \epsilon Z_q^*$ Calculates: $X_a^* = r_{a2}^* P, X_a^{*'} = r_{a2}^* O_a$ $Y_a^* = h_1(ID_a \| X_a^* \| X_a^{*'} \| T_a^*)$		
	$\xrightarrow{ID_a, X_a^*, Y_a^*, T_a^*}$	
		Verifies $T_a^* - T_{cs} \leq \Delta T$ $X_a' = h(ID_a \| r_{cs1}) X_a^*$ $Y_a \stackrel{?}{=} (Y_a = h_1(ID_a \| X_a^* \| X_a' \| T_a^*))$ Selects random number $r_{cs2} \in Z_q^*$ $X_{cs} = r_{cs2} P, T_{S_k} = r_{cs2} X_a^*$ $S_k = h_1(T_{S_k})$ $Y_{cs} = h_1(ID_a \| ID_{cs} \| X_a^* \| X_a' \| T_a^* \| T_{cs} \| T_{S_k})$
	$\xleftarrow{ID_a, X_{cs}, Y_{cs}, T_{cs}}$	
Verifies whether $T_{cs} - T_a \leq \Delta T$ Calculates: $T_{S_k}^* = r_{a1}^* X_{cs}$ Verifies: $Y_{cs} \stackrel{?}{=} (Y_{cs}^{*'} = h_1(ID_a \| ID_{cs} \| X_a^* \| X_a^{*'} \| T_a^* \| T_{cs} \| T_{S_k}^*)$ Establishes the session key $S_k^* = h_1(T_{S_k}^*)$ with $\mathcal{CS}$		

FIGURE 5. Mobile user impersonation attack.

$$Y_{cs} = h_1(ID_a \| ID_{cs} \| X_a^* \| X_a' \| T_a^* \| T_{cs} \| T_{S_k}) \quad (15)$$

Step 6: Afterwards, $\mathcal{CS}$ sends a challenge message $\langle ID_a, X_{cs}, Y_{cs}, T_{cs} \rangle$ back to $\mathcal{U}_A$.

Step 7: On receiving $\langle ID_a, X_{cs}, Y_{cs}, T_{cs} \rangle$, $\mathcal{U}_A$ then calculates following after checking freshness of timestamp T_{cs}:

$$T_{S_k}^* = r_{a1}^* X_{cs}, \quad (16)$$

$$Y_{cs} \stackrel{?}{=} (Y_{cs}^{*'} = h_1(ID_a \| ID_{cs} \| X_a^* \| X_a^{*'} \| T_a^* \| T_{cs} \| T_{S_k}^*)) \quad (17)$$

$$S_k^* = h_1(T_{S_k}^*). \quad (18)$$

Hence, $\mathcal{U}_A$ can successfully impersonate as a legitimate mobile user $\mathcal{MU}_a$for sharing session key with $\mathcal{CS}$. Therefore, the scheme of Tiwari *et al.*is vulnerable to the user impersonation attack. The procedure for mobile user impersonation attack is demonstrated in FIGURE 5.

VII. PROPOSED SCHEME

A brief description of our proposed scheme is described in this section. The proposed scheme consists of the following phases.

A. DATA ACCESS PHASE

The data-access phase provides a method to authenticate the user who is interested in accessing the outsourced data. Data access systems are mostly dependent on password or private data tokens. These systems can be attacked by several security attacks by the attacker for accessing the data maliciously (e.g., denial-of-service, replay, and forgery attacks). In this phase overcomes the attacks present against Tiwari *et al.'s* scheme [10], and an improved data access model is introduced. The FIGURE 6 shows the authentication environment for the proposed scheme in which the user requests the login message to the cloud server on a wireless medium. After receiving the login request, the cloud server verifies the requested message and transmits the challenge message to the user. In the end, the user authenticates the legitimacy of

FIGURE 6. Authentication environment.

Mobile User $\mathcal{MU}_a$		*Cloud Server* $\mathcal{CS}$
Registration Phase:		
$\mathcal{MU}_a$ chooses ID_a and PW_a and imprints BIO_a into $\mathcal{MD}_a$ $\mathcal{MD}_a$ computes: $Gen(BIO_a) = (E_{a1}, E_{a2})$ Produces an arbitrary number r_{a1} Computes: $PWP_a = h_1(ID_a \| PW_a \| E_{a1} \| r_{a1})$		
	$\xrightarrow{ID_a, PWP_a}$	
		$\mathcal{CS}$ selects random number r_{cs1} $O_a = h_1(ID_a \| sk_{cs})$ $P_a = PWP_a \oplus h_1(ID_a \oplus r_{cs1})$ $Q_a = h_1(ID_a \| PWP_a \| h_1(ID_a \oplus r_{cs1}))$ $R_a = h_1(ID_a \oplus PWP_a) \oplus O_a$
	$\xleftarrow{P_a, Q_a, R_a}$	
$\mathcal{MD}_a$ stores $(P_a, Q_a, R_a, r_{a1}, Gen(), Rep())$ into $\mathcal{MD}_a$		

FIGURE 7. Registration phase of the proposed scheme.

the cloud server. After that, the cloud server and user share the session key. In this way, the cloud server and user authenticate and communicate with each other.

Our proposed data access system offers user authentication for secure data access which consists of the following phases:

1) REGISTRATION PHASE

In this phase, mobile user $\mathcal{MU}_a$ registers himself at cloud server $\mathcal{CS}$ by following these steps:

Step 1: At first, for the registration to acquire different cloud services, $\mathcal{MU}_a$ selects an identity ID_a and password PW_a, biometric information BIO_a and submit these information towards $\mathcal{MD}_a$. Then $\mathcal{MD}_a$ computes $Gen(BIO_a) = (E_{a1}, E_{a2})$. Furthermore, $\mathcal{MD}_a$ generates a random number $ra1$ and computes $PWP_a = h_1(ID_a \| PW_a \| E_{a1} \| r_{a1})$. After that, ID_a and PWP_a is sent over a secure channel to $\mathcal{CS}$.

Step 2: After receiving ID_a and PWP_a, $\mathcal{CS}$ generates a random number r_{cs1} and computes $O_a = h_1(ID_a \| sk_{cs})$, $P_a = PWP_a \oplus h_1(ID_a \oplus r_{cs1})$, $Q_a = h_1(ID_a \| PWP_a \| h_1(ID_a \oplus r_{cs1}))$ and $R_a = h_1(ID_a \oplus PWP_a) \oplus O_a$.

Step 3: After that, P_a, Q_a and R_a is sent to $\mathcal{MD}_a$ over a secure channel.

Step 4: After receiving (P_a, Q_a, R_a), $\mathcal{MU}_a$ stores $(P_a, Q_a, R_a, ra1, Gen(\cdot), Rep(\cdot))$ into $\mathcal{MD}_a$. The registration phase is also displayed in FIGURE 7.

2) LOGIN AND AUTHENTICATION PHASE

This phase describes the login and authentication method between the mobile user $\mathcal{MU}_a$ and cloud server $\mathcal{CS}$. The detailed description is given below:

Mobile User $\mathcal{MU}_a$	Cloud Server $\mathcal{CS}$
Login and Authentication Phase:	
$\mathcal{MU}_a$ chooses ID_a and PW_a and imprints BIO_a into $\mathcal{MD}_a$	
$\mathcal{MD}_a$ computes:	
$Rep(BIO'_a, E_{a2}) = (E_{a1})$	
Computes:	
$PWP_a = h_1(ID_a \| PW_a \| E_{a1} \| r_{a1})$	
$h_1(ID_a \oplus r_{cs1}) = P_a \oplus PWP_a$	
$O_a = h_1(ID_a \| PWP_a) \oplus R_a$	
$Q'_a = h_1(ID_a \| PWP_a \| h_1(ID_a \oplus r_{cs1}))$	
Checks $Q'_a \stackrel{?}{=} Q_a$ whether the user registered with $\mathcal{CS}$, if it holds true then	
Obtains the current timestamp T_a selects the random number $r_{a2} \in \mathcal{Z}_q^*$	
Calculates:	
$X_a = r_{a2}P$, $X_a^* = r_{a2}pk_{cs}$	
$Y_a = h_1(ID_a \| X_a \| O_a \| T_a)$	
$PID_a = ID_a \oplus X_a$	
$\xrightarrow{PID_a, X_a^*, Y_a, T_a}$	
	Verifies $T_a - T_{cs} \leq \Delta T$
	$ID_a = PID_a \oplus X_a$
	$O_a = h_1(ID_a \| sk_{cs})$
	$Y_a \stackrel{?}{=} (Y'_a = h_1(ID_a \| X_a \| O_a \| T_a))$
	Selects random number $r_{cs2} \in \mathcal{Z}_q$
	$X_{cs} = r_{cs2}P$, $T_{S_k} = r_{cs2}X_a$
	$S_k = h_1(T_{S_k})$
	$Y_{cs} = h_1(ID_a \| ID_s \| X_a \| X'_a \| T_a \| T_{cs} \| T_{S_k})$
	$\xleftarrow{X_{cs}, Y_{cs}, T_{cs}}$
Verifies $T_{cs} - T_a \leq \Delta T$	
Calculates: $T_{S_k} = r_{a2}X_{cs}$	
Verifies:	
$Y_{cs} \stackrel{?}{=} (Y'_{cs} = h_1(ID_a \| ID_s \| X_a \| X_a^* \| T_a \| T_{cs} \| T_{S_k}))$	
Establish the session key $S_k = h_1(T_{S_k})$ with $\mathcal{CS}$	

FIGURE 8. Login and authentication phase of the proposed scheme.

Step 1: $\mathcal{MU}_a$ enters his/her ID_a and PW_a and BIO_a into $\mathcal{MD}_a$. Then $Rep(BIO'_a, E_{a2}) = (E_{a1})$ is calculated by $\mathcal{MD}_a$. After that, it computes $PWP_a = h_1(ID_a \| PW_a \| E_{a1} \| r_{a1})$, $h_1(ID_a \oplus r_{cs1}) = P_a \oplus PWP_a$, $O_a = h_1(ID_a \| PWP_a) \oplus R_a$ and $Q'_a = h_1(ID_a \| PWP_a \| h_1(ID_a \oplus r_{cs1}))$. $\mathcal{MD}_a$ checks whether $Q_a \stackrel{?}{=} Q'_a$ is true. If the said condition holds true, then $\mathcal{MD}_a$ generates a current time stamp T_a and a random number r_{a2} for determining $X_a = r_{a2}P$, $X_a^* = r_{a2}pk_{cs}$, $Y_a = h_1(ID_a \| X_a \| O_a \| T_a)$ and $PID_a = ID_a \oplus X_a$. After that, a request message $\{PID_a, X_a^*, Y_a, T_a\}$ is forwarded to $\mathcal{CS}$ over a public channel.

Step 2: When $\mathcal{CS}$ receives the request message $\{PID_a, X_a^*, Y_a, T_a\}$, it checks the legitimacy of the timestamp as $T_a - T_{cs} \leq \Delta T$ and determines $ID_a = PID_a \oplus X_a$, $O_a = h_1(ID_a \| sk_{cs})$. After that, $\mathcal{CS}$ checks and calculates $Y_a \stackrel{?}{=} (Y'_a = h_1(ID_a \| X_a \| O_a \| T_a))$, if this authentication check is false then the session will be aborted, otherwise proceeds to the further calculations. Moreover, $\mathcal{CS}$ selects a random number $r_{cs2} \in \mathcal{Z}_q$ and computes $X_{cs} = r_{cs2}P$, $T_{S_k} = r_{cs2}X_a$, $S_k = h_1(T_{S_k})$, $Y_{cs} = h_1(ID_a \| ID_s \| X_a \| X'_a \| T_a \| T_{cs} \| T_{S_k})$. After that, $\mathcal{CS}$ sends a challenge message $\{X_{cs}, Y_{cs}, T_{cs}\}$ to $\mathcal{MU}_a$, so that the $\mathcal{MU}_a$ can check the legitimacy of $\mathcal{CS}$.

Step 3: Upon receiving the challenge message, $\mathcal{MU}_a$ verifies the timestamp using $T_{cs} - T_a \leq \Delta T$. If timestamp is verified to be correct, the $\mathcal{MU}_a$ calculates a session token to validate the session key $T_{S_k} = r_{a2}X_{cs}$. After that, $\mathcal{MU}_a$ authenticates $\mathcal{CS}$ using the equation $Y_{cs} \stackrel{?}{=} (Y'_{cs} = h_1(ID_a \| ID_s \| X_a \| X_a^* \| T_a \| T_{cs} \| T_{S_k}))$. If $\mathcal{CS}$

is a valid server, then the session key $S_k = h_1(T_{S_k})$ between $\mathcal{MU}_a$ and $\mathcal{CS}$ will be established. Otherwise, the session will be aborted. This whole procedure is demonstrated in FIGURE 8.

3) PASSWORD CHANGE PHASE

If the user wants to update his password, our scheme provides the facility to update his/her password. In this method, $\mathcal{MU}_a$ inputs ID_a, PW_a and imprints BIO_a into $\mathcal{MD}_a$. Then, $\mathcal{MD}_a$ calculates $PWP_a = h_1(ID_a \| PW_a \| E_{a1} \| r_{a1})$, $h_1(ID_a \oplus r_{cs1}) = P_a \oplus PWP_a$ and $Q'_a = h_1(ID_a \| PWP_a \| h_1(ID_a \oplus r_{cs1}))$. Moreover, $\mathcal{MD}_a$ checks $Q_a \stackrel{?}{=} Q'_a$. If the said condition is true, then $\mathcal{MU}_a$ is allowed to change the old password PW_a to a new password PW_a^{new}, otherwise the request for password change will be rejected. Using new password, the update to the system is given by $\mathcal{MD}_a$ using $PWP_a^{new} = h_1(ID_a \| PW_a^{new} \| E_{a1} \| r_{a1})$, $h_1(ID_a \oplus r_{cs1}) = P_a^{new} \oplus PWP_a^{new}$ and $Q_a^{new} = h_1(ID_a \| PWP_a^{new} \| h_1(ID_a \oplus r_{cs1}))$ and replaces the credentials of P_a, Q_a, R_a with P_a^{new}, Q_a^{new}, R_a^{new} for successful updation of user's password.

B. DATA STORAGE SYSTEM

$\mathcal{MU}_a$ needs to acquire the cloud storage for sharing his/her encrypted data with a recipient user $\mathcal{MU}_a$. The user $\mathcal{MU}_a$ invokes the data access service during the data access phase in which $\mathcal{MU}_a$ establishes a session key with $\mathcal{CS}$. After that, $\mathcal{MU}_a$ produces the ciphertext of the data, which is saved in a file D_f. For this purpose, $\mathcal{MU}_a$ uses a convergent encryption solution in which the hash function is utilized to derive the shared key through the input of D_f. To share with other users, the encrypted file F'_{en} is stored by the $\mathcal{MU}_a$ in cloud storage. Firstly, authentication is performed for $\mathcal{MU}_a$ to validate its authenticity as a valid user by the server $\mathcal{CS}$. Then upon receiving the file signature from $\mathcal{MU}_a$ the $\mathcal{CS}$ stores the data on the cloud storage. Next, $\mathcal{MU}_a$ computes a proxy re-encryption token, for flexible sharing of outsourcing, so that a valid user can decrypt the data of the file. The following steps perform the data storage procedure:

Step 1: The $\mathcal{MU}_a$ choose a random number and computes the key (key) of file enciphering using symmetric encryption ($SymEnc()$) on the file D_f.

$$key = h_1(D_f), \tag{19}$$

$$f_{en} = SymEnc(D_f, key). \tag{20}$$

Step 2: For flexible sharing of D_f with verified user, first layer encryption is performed by the $\mathcal{MU}_a$

$$Sem = D_f \oplus r_a P_2 \oplus keyP \oplus r_a Pk_{cs}. \tag{21}$$

where Sem is an encrypted file, in which random number r_a is related to $\mathcal{CS}$ and cipher key key is related to the data file D_f.

Step 3: $\mathcal{MU}_a$ calculates the file signature f_{sign} to provide the authenticity proof and to remain the integrity of the D_f throughout the transmission. During this process, $\mathcal{MU}_a$ chooses a random number $x_a \in \mathcal{Z}_p^*$ and computes $x = x_a P$. The following calculations are performed during the generation of sig_{D_f} over encrypted file f_{en}:

$$f_{sig} = h_1(Sem)Sk + x_a h_2(PK_{cs}). \tag{22}$$

Step 4: $\mathcal{MU}_a$ computes a proxy encryption token (A_1, A_2) and it is sent to $\mathcal{CS}$.

$$C_1 = r_1 \oplus h_1(e(r_1 P, PK_{cs}))^{SK_{cs}} \tag{23}$$

$$C_2 = e(r_1 P, PK_{cs}) \tag{24}$$

Step 5: $\mathcal{MU}_a$ generates the message $M_5 = (Sem, C_1, C_2, sig_f, X_a)$ and transmits towards the $\mathcal{CS}$ for the storage of encrypted data.

Step 6: After getting the message M_5 along with file meta information, the authentication is performed for $\mathcal{MU}_a$ by the server $\mathcal{CS}$ and the data is stored on the cloud storage after verifying the following signature operation:

$$Pf_{sig} \stackrel{?}{=} h_1(Sem)PK_{ui} + h_1(PK_{cs})X_a \tag{25}$$

$\mathcal{CS}$ stores meta information and message M_5, if the above authentication is correct, otherwise $\mathcal{CS}$ aborts the request of $\mathcal{MU}_a$ for storing the encrypted file.

C. DATA SHARING SYSTEM

$\mathcal{CS}$ gives a choice to the user $\mathcal{MU}_b$ for searching the file by inserting keywords related to the file. After getting the credentials entered by $\mathcal{MU}_b$, $\mathcal{CS}$ searches the database query to find a match and then outputs the signature and user identity of the file owner. $\mathcal{CS}$ gets the specific file signature and identity and then sends them to $\mathcal{MU}_b$ to access the file. A request query is developed by $\mathcal{CS}$ with the input file D_f and identity $\mathcal{ID}_b$ of recipient user $\mathcal{MU}_b$ who needs to access the data. The developed query is forwarded to $\mathcal{MU}_a$. Then a credential token is generated by $\mathcal{MU}_b$ to grant the permission for decrypting the data file D_f. The credential token helps $\mathcal{MU}_b$ to decrypt the data provided by $\mathcal{CS}$. In this method, before generating the tokens, the identity $\mathcal{ID}_b$ of the requested user $\mathcal{MU}_b$ and file access permission are verified by the owner $\mathcal{MU}_a$ of the data file D_f. Then, $\mathcal{MU}_b$ uses the credential token to extract the symmetric key key to decrypt the data file D_f. After that, the credential token is used to extract the symmetric key by $\mathcal{MU}_b$ for decrypting D_f. The details of the data sharing services are described below:

Step 1: Proxy re-encryption is performed by the $\mathcal{CS}$ using (C_1, C_2) for the provision of flexible access with $\mathcal{MU}_b$ without revealing the privacy of the file D_f. $\mathcal{CS}$ perform above re-encryption process by the following:

1) Calculates

$$A_2^* = (A_2)^{SK_{cs}}. \tag{26}$$

2) Obtains the value r_1 by calculating

$$r_1 = A_1 \oplus h_1(A_2^*). \tag{27}$$

3) Proxy re-encryption is performed on the file D_f by following equation:

$$f_{en}^{\epsilon} = (Sem) \oplus r_1 P \oplus SK_{cs} r_1 P. \tag{28}$$

where f_{en}^{ϵ} is an output of (Sem) by $\mathcal{CS}$ and forward to the $\mathcal{MU}_b$.

Step 2: $\mathcal{CS}$ generates request using input of file identifier and receiver $\mathcal{MU}_b$ who wants to get the data send it towards $\mathcal{MU}_a$. After getting request from $\mathcal{CS}$, $\mathcal{MU}_a$ checks the authorization of the valid and requesting user $\mathcal{MU}_b$. If $\mathcal{MU}_a$ verifies the authenticity and authorization of $\mathcal{MU}_b$ then he calculates the credential tokens by the equation below and send (A_1, A_2) to the requesting user $\mathcal{MU}_b$.

$$A_3 = key \oplus h_1(e(PO_a, PK_{cs}))^{key}, \tag{29}$$

$$A_4 = e(PK_{cs}, PO_a)^k ey\ SK_{cs}. \tag{30}$$

Step 3: The controlled sharing of the data is performed by sending credential token (A_3, A_4) through $\mathcal{MU}_a$ to $\mathcal{CS}$. After extracting the enciphering key, symmetric decryption is performed as mentioned below to avail the file D_f.

1) Calculates

$$A_4^* = (A_4)^{\frac{1}{S}K_{cs}}, key =_A 3 \oplus h_1()A_4''. \tag{31}$$

2) Decryption process is performed to derive D_f.

$$f_{en} = Sem \oplus keyP. \tag{32}$$

3) $SymEnc()$ algorithm is executed to decrypt the file D_f upon cipher text f_{en}'s input. Then, secret key (key) is shared by the following equation:

$$D_f = SymEnc(f_{en}), key). \tag{33}$$

4) In order to verify the authenticity of $\mathcal{MU}_a$, secret key (key) and integrity of file D_f, the following action is performed by the user $\mathcal{MU}_b$:

$$key \stackrel{?}{=} h_1(D_f). \tag{34}$$

If the above equations are correct then $\mathcal{MU}_b$ gets the file f and executes the operations of file.

VIII. SECURITY ANALYSIS OF PROPOSED SCHEME

In this section, we present the analysis of security according to an informal and formal way. We have performed the security analysis to demonstrate that our protocol provides sufficient protection against the major security threats.

A. INFORMAL SECURITY ANALYSIS

Here, we informally analyze the security of the proposed protocol. The analysis of security demonstrates that the introduced scheme is correct, secure and efficient against the numerous attacks.

1) MUTUAL AUTHENTICATION

The $\mathcal{CS}$ can authenticate $\mathcal{MU}_a$ by verifying $Y_a \stackrel{?}{=} h_1(ID_a \| X_a \| O_a \| T_a)$. As, the calculation of Y_a needs the valid calculation of $O_a = h_1(ID_a \| sk_{cs})$, which involves the private key of $\mathcal{CS}$. Moreover, $\mathcal{MU}_a$ can also authenticate $\mathcal{CS}$ by verifying $Y_{cs} \stackrel{?}{=} (Y'_{cs} = h_1(ID_a \| ID_s \| X_a \| X'_a \| T_a \| T_{cs} \| T_{sk}))$. Note that ID_a and PW_a of $\mathcal{MU}_a$ are required to calculate Y_{cs} and these values are unknown to $\mathcal{U}_A$. Thus, the mutual authentication between $\mathcal{MU}_a$ and $\mathcal{CS}$ is offered by our scheme.

2) CLOUD SERVER IMPERSONATION ATTACK

An adversary $\mathcal{U}_A$ cannot impersonate the cloud server $\mathcal{CS}$ even he/she intercepts the challenge message $\{X_{cs}, Y_{cs}, T_{cs}\}$. Because the calculation of $Y_{cs} = h1(ID_a \| ID_s \| X_a \| X'_a \| T_a \| T_{cs} \| T_{sk})$ requires sk_{cs} of the $\mathcal{CS}$.

3) MOBILE USER IMPERSONATION ATTACK

An attacker $\mathcal{U}_A$ cannot masquerade a legal user $\mathcal{MU}_a$ even if he/she intercepts the login request message $\{PID_a, X_a^*, Y_a, T_a\}$. Because the calculation of $Y_a = h_1(ID_a \| X_a \| O_a \| T_a)$ needs the valid calculation of O_a. To calculate O_a, PWP_a is needed, which is unknown to $\mathcal{U}_A$. So, our introduced scheme provides resistance against user impersonation attack.

4) VIOLATION OF USER ANONYMITY

The protocol does not send identity ID_a of $\mathcal{MU}_a$ in plain text. However, $PID_a = ID_a \oplus X_a$ is calculated and sent over a private channel to $\mathcal{CS}$. Moreover, the ID_a can only be derived by the authorized $\mathcal{CS}$ by utilizing the private key of $\mathcal{CS}$. So, user anonymity is offered by our protocol.

5) MAN-IN-THE-MIDDLE ATTACK

Suppose the message $\{PID_a, X_a^*, Y_a, T_a\}$ is intercepted by $\mathcal{U}_A$. But, still the login information can not be changed by $\mathcal{U}_A$ due to the session specific PID_a. Furthermore, the session-specific arbitrary number is needed to determine X_a that is utilized for calculating Y_a. So, the man-in-the-middle attack is efficiently resisted by the proposed protocol.

6) MOBILE DEVICE STOLEN ATTACK

The mobile device consists of $\{P_a, Q_a, R_a, r_{a1}, Gen(\cdot), Rep(\cdot)\}$. The $\mathcal{MD}_a$ has no such parameter that can support $\mathcal{U}_A$ to guess the $\mathcal{MU}'_a s$ confidential parameter. So, if $\mathcal{U}_A$ gets the $\mathcal{MD}_a$ of $\mathcal{MU}_a$ even then he can not get any benefit from the stored values in $\mathcal{MD}_a$. Therefore, $\mathcal{U}_A$ can not do mobile device stolen attack.

7) PERFECT FORWARD SECRECY

$\mathcal{MU}_a$ and $\mathcal{CS}$ calculates the Y_a and Y_{cs} enclosed by r_{a2} and r_{cs2} which are session-explicit arbitrary numbers. Therefore, if $\mathcal{U}_A$ derives the long term secret key sk_{cs} of any user, even then the calculation of preceding session keys can not be done by $\mathcal{U}_A$. Thus, the proposed scheme offers the perfect forward secrecy.

8) STOLEN VERIFIER AND PRIVILEGED INSIDER ATTACK

Our protocol does not maintain any database and $\mathcal{MU}_a$ is validated by utilizing the $\mathcal{CS}'s$ private key. Similarly ID_a and PW_a are not sent to $\mathcal{CS}$ in plain-text through a public channel. So our scheme resists against privileged insider and stolen verifier attack.

9) DATA CONFIDENTIALITY

Our protocol provides the ciphertext to store on the cloud storage. Particularly, by using the symmetric encryption, data files are encrypted and the private key is shared with the user by using the asymmetric key protocol. The hardness of protecting the confidentiality of encrypted data based on the related symmetric key algorithm's security and the privacy of the randomly selected data key and secret key.

10) NO REPUDIATION

In the data access mechanism, $\mathcal{MU}_a$ has one secret credential to identify itself uniquely, where PW_a is a combination of character, numeric, and special symbols. Proof of service origin and proof of receipt service is provided in our protocol by using the user's secret credentials to share the common session key between the involved entities. So, the non-repudiation is offered by the data access mechanism. Instead, signature S_f on the encrypted data, S_{em} is generated by $\mathcal{DO}$ on the decipher data for the operation of data uploading.

11) FLEXIBLE DATA ACCESS CONTROL

$\mathcal{DO}$ produces the token of proxy encryption $\langle A_1, A_2\rangle$ and transmit it to the $\mathcal{CS}$. These tokens can be used by $\mathcal{CS}$ to convert the encrypted text into another encrypted text so that legitimate receipt can decrypt the text. If an illegal person gets access to the re-encrypted message and try to get the secret key sk_{cs} from the secret credentials; still, she/he is not able to get the value of sk_{cs}.

B. FORMAL SECURITY ANALYSIS

To perform a formal security analysis of our scheme, we have used the Random Oracle Model (ROM). For the sake of simplicity, we have selected the security model presented in [29] as our security model for the proposed scheme. We will provide detailed security proof and privacy proof of the proposed scheme. The security and privacy proof of our security model is similar to [29], but the only difference is that our scheme is ECC based.

Theorem T1: Suppose that P_{rp} indicates the presented protocol. $\mathcal{D}$ denotes the password dictionary, and Zipf's law is being followed for its frequency distribution. $\mathcal{U}_A$ denotes adversary, who can make q_s number of send oracle queries with execution time t_m. While $Adv^{AKE}_{P_{rp},\mathcal{U}_A}$ indicates the adversary $\mathcal{U}_A$ in breaking AKE security of P_{rp}. By using the assumption of CDH problem, it is obvious that if the hash function $h(\cdot)$ acts like random oracle and the scheme P_{rp} is unbreakable against various attacks then

$$Advt^{AKE}_{P_{rp},D}(\mathcal{U}_A) \leq M' \cdot q_s^{N'} + \epsilon(w)$$

where M' and N' are the parameters of Zipf's, w is the security parameter and ϵ is an negligible function.

Proof: Multiple games from $Game_0$ to $Game_6$ have been used to prove this theorem. In every game, $\mathcal{U}_A$ will use *Test* query to guess the correct bit and it is offered as S_i and $P_r[S_i]$ is the corresponding probability.

Game 0: Under the random oracle model, this game is taken as real attack scenario. As per the definition of $\mathcal{U}_A$ advantage [29], we achieved

$$Advt^{AKE}_{P_{rp},D}(\mathcal{U}_A) = P_r[S_0] \tag{35}$$

Game 1: $Game_1$ purely simulates the one-way secure hash function $h(\cdot)$ by establishing a hash list L_{hs}. Furthermore, *Send*, *Test*, *Execute*, *Reveal*, and *Corrupt* queries are also simulated. This game is not different from $Game_0$. So, we obtain the following equation:

$$|P_r[S_1] - P_r[S_0]| \leq \epsilon(w) \tag{36}$$

Game 2: In $Game_2$, all those sessions in which collision occurs during the simulation of hash functions have been ruled out for the transcript $\{PID_a, X_a^*, Y_a, T_a, X_{cs}, Y_{cs}, T_{cs}\}$. This game will be aborted if some collision occurs. As per the birthday paradox, we got

$$|P_r[S_2] - P_r[S_1]| \leq \epsilon(w) \tag{37}$$

GAME 3: With the help of *Execute* query, the simulation rules of all sessions have been modified in this game. For the calculation of private session key, we have used private hash function $\acute{h}(\cdot)$ instead of actual hash function $h(\cdot)$. Furthermore, the collision of hash function and transcripts have been removed from this game. Adversary $\mathcal{U}_A$ has capability to distinguish between $Game_2$ and $Game_3$ if and only if she/he can successfully calculate X_{cs} in passive session. But the breaking of hard problem CDH is not feasible. We have used self reducibility of CDH hard problem to integrate (X_a^*, y) to the passive session. So, to do the required task, we select some numbers randomly r_{a1}, r_{cs1}, r_{a2} and r_{cs2} and compute $T_{Sk} = r_{cs2}X_a$ and $T_{Sk} = r_{a2}X_{cs}$. $\mathcal{U}_A$ can make a query X_{cs}, Y_{cs}, T_{cs} to hash oracle, if she/he discriminate $Game_2$ and $Game_3$. Under the difficulty of hard problem CDH, we got:

$$|P_r[S_3] - P_r[S_2]| \leq \epsilon(w) \tag{38}$$

GAME 4: In $Game_4$, we initiate to manage the active session for *Send* query. This rule is considered in this game that $\mathcal{U}_A$ can calculate the valid X_{cs} to masquerade $\mathcal{MU}_a$. This rule is charged to somehow as follow: Calculate $Y_{cs} = h_1(ID_a\|ID_s\|X_a\|X'_a\|T_a\|T_{cs}\|T_{sk})$ and determines $Y'_{cs} \stackrel{?}{=} Y_{cs}$. If it holds true then $\mathcal{CS}$ looks forwards for record $\{PID_a, X_a^*, Y_a, T_a\}$ presented in L_{hs}. The game will be aborted if a record is found. Y_{cs} in the presented scheme is unbearable due to the hardness of CDH. So, we got:

$$|P_r[S_4] - P_r[S_3]| \leq \epsilon(w) \tag{39}$$

GAME 5: The active session for *Send* query ($\mathcal{MU}_a$ $\{X_{cs}, Y_{cs}, T_{cs}\}$) is continued in $Game_5$. This game is also expressed by aborting the game with given rule, where $\mathcal{U}_A$ can guess $X_{cs} = r_{cs2}P$ to impersonate $\mathcal{CS}$ without knowing the hash query $h(\cdot)$. In order to accomplish this goal, the rule for the queries is changed as follows: The game will be aborted if $\mathcal{U}_A$ successfully find the record $\{X_{cs}, Y_{cs}, T_{cs}\}$ in L_{hs}. Otherwise, the session key will be generated as $S_k = h_1(T_{S_k})$. $\mathcal{U}_A$ will win this game if correctly guesses T_{S_k}. Similarly to $Game_4$, we got the following:

$$|P_r[S_5] - P_r[S_4]| \leq \epsilon(w) \quad (40)$$

GAME 6: In this game, the session key S_k of $\mathcal{MU}_a$ and $\mathcal{CS}$ is randomly selected, while the advantage of $\mathcal{U}_a$ is negligible to guess S_k. There is only one way for $\mathcal{U}_A$ to win this game is to know the password of $\mathcal{MU}_a$. But $\mathcal{U}_A$ cannot know the password from $Game_6$. According to zipf's law, we got the following:

$$|P_r[S_6]| \leq M' \cdot q_s^{N'} \quad (41)$$

After adding the Equations (35) - (41), we got

$$Advt^{AKE}_{P_{rp},D}(\mathcal{U}_A) \leq M' \cdot q_s^{N'} + \epsilon(w)$$

Theorem T2: Suppose that P_{rp} is the presented protocol and $\mathcal{U}_A$ is an adversary breaking the anonymity of the mobile user $\mathcal{MU}_a$. Then the advantage of $\mathcal{U}_A$ to break the anonymity of $\mathcal{MU}_a$ is bounded by

$$Advt^{AKE}_{P_{rp}}(\mathcal{U}_A) \leq \epsilon(w)$$

Proof: It is assumed that $\mathcal{U}_A$ can easily violate that anonymity of P_{rp} with negligible advantage. This aim is achieved by implementing $\mathcal{U}_A$ to make an algorithm to trash the *CDH* problem. The description of the algorithm is given as follows.

Choose $r_{a1}, r_{a2}, r_{cs1}, r_{cs2} \in \mathcal{Z}_q^*$ and input two records (P, $r_{a2}P$, $r_{a2}pk_{cs}$) and (r_{cs}, P, $r_{cs2}P$, sk_{cs}) where sk_{cs} is the private key of $\mathcal{CS}$.

- Let $\mathcal{MU}_a$ is a valid user having his/her own password and mobile device.
- Let $X_{ua} = r_{a2}P$, $X^*_{us} = r_{a2}pk_{cs}$ and simulates the required procedure according to the definition of the proposed protocol with the cloud server $\mathcal{CS}$. cid^c_C is used as session identifier for this simulation.
- Let $X_{ua} = r_{a2}P$ and $X^*_{ua} = r_{a2}pk_{cs}$ and simulate the method according to the definition of the proposed protocol. It is quite possible that $\mathcal{CS}$ can reject the message coming from $\mathcal{MU}_a$. In this case, $\mathcal{MU}_a$ can choose two random bit strings X_{cs} and Y_{cs}.
- Choose $r_{a2} \in \mathcal{Z}_q^*$, let $X_{ua} = r_{a2}P$ and $X^*_{ua} = r'_{a2}pk_{cs}$, and simulates the method with server $\mathcal{CS}$ using X_{ua} and X^*_{ua}. In this case, session identifier is denoted as cid^K_c.
- Two queries **TestAnon**(cid^c_C, cid^j_C) and **TestAnon**(cid^c_C, cid^k_C) are generated by $\mathcal{U}_A$, while the returned bits are bt_1 and bt_2.
- If $bt_1 = 1$ and $bt_2 = 0$, then output (P, $r_{cs1}P$, Sk_{cs}, r_{cs2}) is considered a Diffie-Hellman tuple. If $bt_1 = 0$ and $bt_2 = 0$, then none will be a Deffie-Hellmen tuple. If $bt_1 = 0$ and $bt_2 = 1$, then (P, $r_{cs2}P$, r_{cs1}) is a Deffie-Hellman tuple. If $bt_1 = 1$ and $bt_2 = 0$, then both are Diffie-Hellman tuple.

It is possible that the above-said algorithm can be successfully performed within polynomial time. On the basis of self-reducibility of *CDH* problem, we got $Advt_{CDH,P_{rp}}(C)$ $= |\mathbf{Pr}[C(P,\ r_{a2}P,\ pk_{cs},\ r_{a2}pk_{cs}) = 1] - \mathbf{Pr}[C(P,\ r_{cs},\ r_{cs2}P,\ sk_{cs}) = 1]|$, where sk_{cs} is a fixed value. So, we have $Adv_{CDH,P_{rp}}(C) \geq |Pr[\mathbf{TestAnon}(cid^c_C,\ cid^j_c) = 1] - \Pr[\mathbf{TestAnon}(cid^c_C, cid^k_C) = 1]|$. So, it is possible that $\mathcal{U}_A$ can break the anonymity of $\mathcal{MU}_a$ by solving the *CDH* problem, but, it is believed that *CDH* is computationally infeasible with in polynomial time. Hence, the theorem is proved.

TABLE 2. Specifications for implementation.

Items	Specifications
Hardware	core i7
Processor	3.60 GHz
Windows	Windows 10
Language	python
Online cloud server	PythonAnywhere

TABLE 3. Computational overhead.

Protocols	Operations Performed	Computation Cost(ms)
Proposed	$16T_h + 5T_{pm}$	0.0282 ms
Tiwari et al.[10]	$17T_h + 7T_{pm}$	0.0349 ms
Amin et al.[30]	$23T_h$	0.0191 ms
Zhou et al.[31]	$38T_h$	0.0316 ms
Mo et al.[32]	$13T_h + 6T_{pm}$	0.0287 ms

IX. THEORETICAL ANALYSIS

In this section, we have analyzed the performance of our enhanced scheme with numerous relevant schemes [10], [30]–[32] in terms of computational complexity, storage complexity, and communication complexity. The protocols have been presented either considering the proxy re-encryption concept or authenticated data access structure or attribute-based encryption (ABE) for flexible sharing of data in the cloud server. For the analysis, we have chosen the parameters used to perform cryptographic operations described as (i) hash operation and (ii) point multiplication. For evaluating the performance, we implemented the operations performed on the user side using python language on a system equipped with core i7 having 3.60 GHz frequency and configured with Windows 10, whereas the operation performed at the server $\mathcal{CS}$ are implemented on PythonAnywhere, which is an online cloud server. These specifications are also shown in TABLE 2.

A. COMPUTATIONAL COMPLEXITY

In this subsection, we have calculated the computational complexity of our scheme and other related schemes in terms of used cryptographic operations. For calculating the computational cost, we have considered hash T_h and point multiplication T_{pm} functions included in the system of data access, sharing, storage. TABLE 3 shows the number of operations performed in the schemes along with their computation time in milliseconds (ms). The time needed for executing the hash operation is 0.000832 ms whereas, the time required for computing point multiplication is 0.002975 ms. Thus, TABLE 3 displays that our protocol incurs less computational

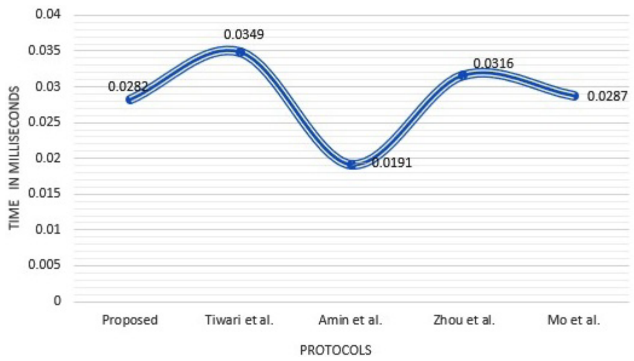

FIGURE 9. Comparison of computational overhead.

TABLE 4. Storage overhead.

Protocols	Storage Cost(in bits)
Proposed	928
Tiwari et al.[10]	928
Amin et al.[30]	1696
Zhou et al.[31]	1088
Mo et al.[32]	1248

cost as compared to many of the related protocols. FIGURE 9 is also showing the comparison of the computational cost of proposed and related schemes graphically. Where, Y-axis represents the computation cost and X-axis represents the protocols.

B. STORAGE OVERHEAD

The storage cost represents the amount of space required to store parameters used by participants. We compared our scheme's storage cost with the related schemes [10], [30]–[32] to find the storage overhead. To calculate the storage cost for the $\mathcal{MU}_a$ and $\mathcal{CS}$, we have considered the following values: 160 bits are reserved for timestamp, user identity, password, XOR, random number and elliptic curve point, whereas, 256 bits are required for the hash digest.

The TABLE 4 demonstrates that our scheme has these same cost as Tiwari *et al.* [10] scheme but our scheme incurs less cost than the rest of the schemes [30]–[32]. It is graphically presented in FIGURE 10 that our scheme is efficient in terms of storage overhead.

C. COMMUNICATION OVERHEAD

In this subsection, the communication cost required for the invocation of the data storage, data access, and data sharing system is calculated. The communication cost is computed in terms of the bit size of the message shared by the participating entities as described below:

1) By establishing a session between $\mathcal{MU}_a$ and $\mathcal{CS}$, they both authenticate each other mutually during data access services.
2) In data sharing services, for controlling the access capability of the $\mathcal{MU}_a$, the $\mathcal{DO}$ engenders the credential token and sends it to the user.

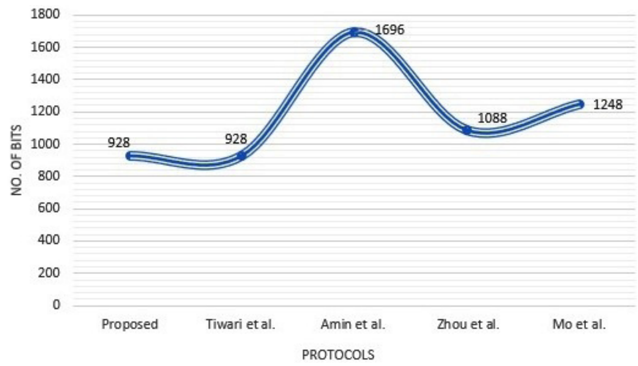

FIGURE 10. Comparison of storage overhead.

TABLE 5. Communication overhead.

Protocols	Communication Cost(in bits)
Proposed	2592
Tiwari et al.[10]	2656
Amin et al.[30]	6688
Zhou et al.[31]	6848
Mo et al.[32]	2912

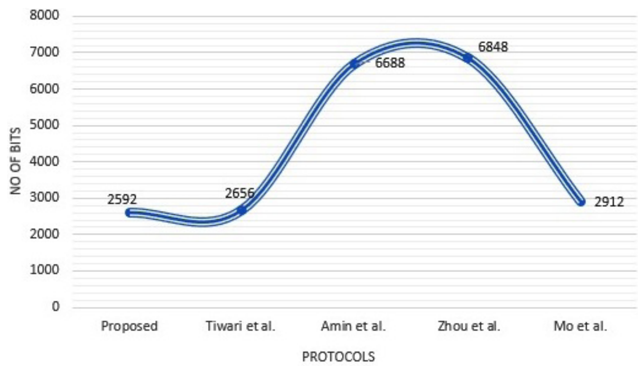

FIGURE 11. Comparison of communication overhead.

TABLE 6. Comparison of security features.

Features	[10]	[30]	[31]	[32]	Proposed
Mutual Authentication	✓	✓	✓	✓	✓
User Impersonation	✗	✓	✓	✓	✓
Server Impersonation	✗	✓	✓	✓	✓
Provide User Anonymity	✗	✗	✓	✗	✓
Man-in-Middle Attack	✓	✓	✓	✓	✓
Mobile Device Stolen Attack	✓	✗	N/A	✗	✓
Perfect Forward Secrecy	✓	✓	✓	✓	✓
Stolen Verifier and Privileged Insider Attack	✓	✓	✓	✓	✓
Data Confidentiality	✓	N/A	N/A	✗	✓
Non-Repudiation	✓	N/A	N/A	N/A	✓
Flexible Data Access Control	✓	N/A	N/A	N/A	✓
Password Guessing Attack	✓	✗	✓	✓	✓

3) In data storage services, to offer flexible and convenient services of data sharing to $\mathcal{MU}_a$, the encrypted data and proxy re-encryption key in cloud storage are uploaded by the $\mathcal{DO}$.

It is evident from TABLE 5 that our protocol requires less number of bits for communication than all of the related protocols [10], [30]–[32]. Whereas, FIGURE 11 also depicts the same communication overhead comparison of the protocols graphically, which states that our protocol is efficient as compared to related protocols.

After analyzing storage, computation, communication costs, and TABLE 6, we can conclude that our scheme is more convenient in terms of resource utilization; also, it provides enhanced security features in comparison to the related protocols. Thus, we can state that our proposed protocol is more efficient than related protocols.

X. CONCLUSION

In this article, we have cryptanalyzed Tiwari et al.'s protocol, which is proposed for the cloud storage environment. Their protocol violates user anonymity because an attacker can easily find the identity of a legitimate user. Furthermore, the protocol presented by Tiwari et al., is also vulnerable to mobile user and cloud server impersonation attacks. Therefore, we propose an improved protocol to resist the said attacks present in Tiwari et al.'s protocol. We used the Random Oracle Model (ROM) for formally analyzing the security of the proposed protocol; an informal analysis of security is also given to show the robustness of the proposed protocol. Moreover, we compared computation, communication, and storage cost of our protocol with related protocols, which shows that our protocol is efficient as compared to relevant protocols.

REFERENCES

[1] X. Wang, L. T. Yang, J. Feng, X. Chen, and M. J. Deen, "A tensor-based big service framework for enhanced living environments," *IEEE Cloud Comput.*, vol. 3, no. 6, pp. 36–43, Nov. 2016.

[2] J. Zeng, L. T. Yang, H. Ning, and J. Ma, "A systematic methodology for augmenting quality of experience in smart space design," *IEEE Wireless Commun.*, vol. 22, no. 4, pp. 81–87, Aug. 2015.

[3] H. Li, K. Ota, M. Dong, and M. Guo, "Mobile crowdsensing in software defined opportunistic networks," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 140–145, Jun. 2017.

[4] A. Singh and K. Chatterjee, "Cloud security issues and challenges: A survey," *J. Netw. Comput. Appl.*, vol. 79, pp. 88–115, Feb. 2017.

[5] B. Libert and D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy re-encryption," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1786–1802, Mar. 2011.

[6] J. Katz and M. Yung, *Applied Cryptography and Network Security: 5th International Conference, ACNS 2007, Zhuhai, China, June 5-8, 2007, Proceedings*, vol. 4521. Berlin, Germany: Springer, 2007.

[7] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 7, pp. 1214–1221, Jul. 2011.

[8] W. Li, K. Xue, Y. Xue, and J. Hong, "TMACS: A robust and verifiable threshold multi-authority access control system in public cloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 5, pp. 1484–1496, May 2016.

[9] C. Wang, Z.-G. Qin, J. Peng, and J. Wang, "A novel encryption scheme for data deduplication system," in *Proc. Int. Conf. Commun., Circuits Syst. (ICCCAS)*, Jul. 2010, pp. 265–269.

[10] D. Tiwari, G. K. Chaturvedi, and G. R. Gangadharan, "ACDAS: Authenticated controlled data access and sharing scheme for cloud storage," *Int. J. Commun. Syst.*, vol. 32, no. 15, p. e4072, Aug. 2019.

[11] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 9, pp. 2546–2559, Sep. 2016.

[12] M. Sookhak, "Dynamic remote data auditing for securing big data storage in cloud computing," Ph.D. dissertation, Univ. Malaya, Kuala Lumpur, Malaysia, 2015.

[13] D. Thilakanathan, S. Chen, S. Nepal, and R. A. Calvo, "Secure data sharing in the cloud," in *Security, Privacy and Trust in Cloud Systems*. Berlin, Germany: Springer, 2014, pp. 45–72.

[14] J. Li, J. Li, Z. Liu, and C. Jia, "Enabling efficient and secure data sharing in cloud computing," *Concurrency Comput., Pract. Exper.*, vol. 26, no. 5, pp. 1052–1066, 2014.

[15] Y. Chen, L. Song, and G. Yang, "Attribute-based access control for multi-authority systems with constant size ciphertext in cloud computing," *China Commun*, vol. 13, no. 2, pp. 146–162, 2016.

[16] Q. Li, J. Ma, R. Li, X. Liu, J. Xiong, and D. Chen, "Secure, efficient and revocable multi-authority access control system in cloud storage," *Comput. Secur.*, vol. 59, pp. 45–59, Jun. 2016.

[17] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Berlin, Germany: Springer, 2007, pp. 288–306.

[18] Q. Liu, G. Wang, and J. Wu, "Time-based proxy re-encryption scheme for secure data sharing in a cloud environment," *Inf. Sci.*, vol. 258, pp. 355–370, Feb. 2014.

[19] J. Zhang and Z. Zhang, "Secure and efficient data-sharing in clouds," *Concurrency Comput., Pract. Exper.*, vol. 27, no. 8, pp. 2125–2143, Oct. 2014.

[20] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2010, pp. 136–149.

[21] G. Zhao, C. Rong, J. Li, F. Zhang, and Y. Tang, "Trusted data sharing over untrusted cloud storage providers," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci.*, Nov. 2010, pp. 97–103.

[22] A. G. Kumbhare, Y. Simmhan, and V. Prasanna, "Designing a secure storage repository for sharing scientific datasets using public clouds," in *Proc. 2nd Int. Workshop Data Intensive Comput. Clouds (DataCloud-SC)*, 2011, pp. 31–40.

[23] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 321–334.

[24] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2011, pp. 53–70.

[25] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Proc. Int. Colloq. Automata, Lang., Program.* Berlin, Germany: Springer, 2008, pp. 579–591.

[26] N. H. Sultan, F. A. Barbhuiya, and M. Laurent, "ICAuth: A secure and scalable owner delegated inter-cloud authorization," *Future Gener. Comput. Syst.*, vol. 88, pp. 319–332, Nov. 2018.

[27] V. S. Miller, "Use of elliptic curves in cryptography," in *Proc. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1985, pp. 417–426.

[28] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[29] M. Karuppiah, A. K. Das, X. Li, S. Kumari, F. Wu, S. A. Chaudhry, and R. Niranchana, "Secure remote user mutual authentication scheme with key agreement for cloud environment," *Mobile Netw. Appl.*, vol. 24, no. 3, pp. 1046–1062, May 2018.

[30] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment," *Future Gener. Comput. Syst.*, vol. 78, pp. 1005–1019, Jan. 2018.

[31] L. Zhou, X. Li, K.-H. Yeh, C. Su, and W. Chiu, "Lightweight IoT-based authentication scheme in cloud computing circumstance," *Future Gener. Comput. Syst.*, vol. 91, pp. 244–251, Feb. 2019.

[32] J. Mo, Z. Hu, H. Chen, and W. Shen, "An efficient and provably secure anonymous user authentication and key agreement for mobile cloud computing," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–12, Feb. 2019.

ZAHID GHAFFAR received the B.S. degree in computer science from the University of Agriculture Faisalabad, Pakistan, in 2018. He is currently pursuing the M.S. degree in computer science from COMSATS University Islamabad at Sahiwal Campus, Pakistan. His research interests include cloud computing, network security, and authenticated key agreement schemes.

SHAFIQ AHMED is currently pursuing the M.S. degree in computer science and the M.C.S. degree from COMSATS University Islamabad at Sahiwal Campus, Pakistan in 2017. He was awarded the Silver medal. His research interests include network security, healthcare authentication, and authenticated key agreement scheme.

KHALID MAHMOOD received the M.S. degree in computer science from Riphah International University, Islamabad, Pakistan, in 2010, and the Ph.D. degree in computer science from International Islamic University, Islamabad, Pakistan, in 2018. He is currently working with COMSATS University Islamabad at Sahiwal Campus. The title of his Ph.D. dissertation is Secure Authenticated Key Agreement Schemes for Smart Grid Communication in Power Sector. His research interests include lightweight cryptography, smart grid authentication, authenticated key agreement schemes, design and development of lightweight authentication protocols using lightweight cryptographic solutions for diverse infrastructures or systems like vehicular ad hoc networks, smart grids, and telecare medical information systems (TMIS).

SK HAFIZUL ISLAM (Senior Member, IEEE) received the M.Sc. degree in applied mathematics from Vidyasagar University, Midnapore, India, in 2006, and the M.Tech. degree in computer applications and the Ph.D. degree in computer science and engineering from the Indian Institute of Technology [IIT (ISM)] Dhanbad, India, in 2009 and 2013, respectively, through the INSPIRE Fellowship Ph.D. Program (funded by the Department of Science and Technology, Government of India). He is currently an Assistant Professor with the Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani (IIIT Kalyani), Kalyani, India. Before joining the IIIT Kalyani, he was an Assistant Professor with the Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani (BITS Pilani), Pilani, India. He has more than six years of teaching and nine years of research experience. He has authored or coauthored 90 research papers in internationally reputed journals and conference proceedings. His research interests include cryptography, information security, WSNs, the IoT, and cloud computing. He is an Associate Editor of the *International Journal of Communication Systems* (Wiley), *Security and Privacy*, and IEEE ACCESS. He was a Reviewer of many reputed international journals and conferences. He was a recipient of the University Gold Medal, the S. D. Singha Memorial Endowment Gold Medal, and the Sabitri Parya Memorial Endowment Gold Medal from Vidyasagar University, in 2006. He was also the recipient of the University Gold Medal from [IIT (ISM)] Dhanbad, in 2009, and the OPERA award from BITS Pilani, in 2015. He is a Senior Member of the ACM.

MOHAMMAD MEHEDI HASSAN (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in February 2011. He is currently an Associate Professor with the Department of Information Systems, College of Computer and Information Sciences (CCIS), King Saud University (KSU), Riyadh, Saudi Arabia. He has authored or coauthored around 180+ publications including refereed IEEE/ACM/Springer/Elsevier journals, conference papers, books, and book chapters. His four publications have been recognized as ESI highly-cited papers. His research interests include cloud computing, edge computing, the Internet of Things, body sensor networks, big data, deep learning, mobile cloud, smart computing, wireless sensor networks, 5G networks, and social networks. He has served as the Chair and a Technical Program Committee Member of numerous reputed international conferences/workshops such as the IEEE CCNC, ACM BodyNets, and the IEEE HPCC. He was a recipient of a number of awards including the Best Journal Paper Award for the IEEE SYSTEMS JOURNAL, in 2018, the Best Paper Award from CloudComp conference, in 2014, and the Excellence in Research Award from King Saud University (two times in row, in 2015 and 2016).

GIANCARLO FORTINO (Senior Member, IEEE) received the Ph.D. degree in computer engineering from the University of Calabria (Unical), Italy, in 2000. He is currently a Full Professor of computer engineering with the Department of Informatics, Modeling, Electronics, and Systems, Unical. He is also a Distinguished Professor at the Wuhan University of Technology and Huazhong Agricultural University (China), a high-end expert at the Huazhong University of Science and Technology (HUST), China, a Senior Research Fellow at the ICAR-CNR Institute, and a CAS PIFI Visiting Scientist at the Shenzhen Institutes of Advanced Technology (SIAT), Shenzhen. He is the Director of the SPEME Laboratory, Unical, as well as the Co-Chair of Joint labs on IoT established between Unical and WUT and SMU and HZAU Chinese universities, respectively. He is the Co-Founder and CEO of SenSysCal S.r.l., a Unical spin-off focused on innovative IoT systems. His research interests include agent-based computing, wireless (body) sensor networks, and the IoT. He is the author of over 430 papers in International journals, conferences, and books. He is the (Founding) Series Editor of the IEEE Press Book Series on Human-Machine Systems and the EiC of the *Internet of Things* (Springer) series. He is currently a member of the IEEE SMCS BoG and the IEEE Press BoG, and the Chair of the IEEE SMCS Italian Chapter. He is also an Associate Editor (AE) of many International journals such as the IEEE TRANSACTIONS ON AUTOMATIC CONTROL (TAC), the IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS (THMS), the IEEE INTERNET OF THINGS JOURNAL (IoTJ), the IEEE SYSTEMS JOURNAL (SJ), the IEEE SERIAL MULTI-METHOD COMBINED MINING (SMCM), *Information Fusion*, JNCA, and EAAI.