IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 30 January 2023, accepted 20 February 2023, date of publication 24 February 2023, date of current version 22 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3249116

RESEARCH ARTICLE

An Effective Privacy-Preserving Blockchain-Assisted Security Protocol for Cloud-Based Digital Twin Environment

GARIMA THAKUR[1], PANKAJ KUMAR[1], DEEPIKA[1], SRINIVAS JANGIRALA[2], (Member, IEEE), ASHOK KUMAR DAS[3], (Senior Member, IEEE), AND YOUNGHO PARK[4], (Member, IEEE)

[1]Srinivasa Ramanujan Department of Mathematics, Central University of Himachal Pradesh, Dharamshala 176206, India
[2]Jindal Global Business School, O. P. Jindal Global University, Sonipat, Haryana 131001, India
[3]Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India
[4]School of Electronics Engineering, Kyungpook National University, Daegu 41566, Republic of Korea

Corresponding authors: Ashok Kumar Das (iitkgp.akdas@gmail.com) and Youngho Park (parkyh@knu.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1I1A3058605.

ABSTRACT Recently, the Digital Twin (DT) technology has procured a lot of attention because of its applicability in the manufacturing and space industries. The DT environment involves the formation of a clone of the tangible object to perform simulations in the virtual space. The combination of conceptual development, predictive maintenance, real-time monitoring, and simulation characteristics of DT has increased the utilization of DT in different scenarios, such as medical environments, healthcare, manufacturing industries, aerospace, etc. However, these utilizations have also brought serious security pitfalls in DT deployment. Towards this, several authentication protocols with different security and privacy features for DT environments have been proposed. In this article, we first review a recently proposed two-factor authentication protocol for DT environments that utilizes the blockchain technology. However, the analyzed scheme is unable to offer the desirable security and cannot withstand various security attacks like offline password-guessing attack, smart card stolen attack, anonymity property, and known session-specific temporary information attack. We also demonstrate that an attacker can impersonate the analyzed protocol's legal user, owner, and cloud server. To mitigate these security loopholes, we devise an effective three-factor privacy-preserving authentication scheme for DT environments. The proposed work is demonstrated to be secure by performing the informal security analysis, the formal security analysis using the widely recognized Burrows-Abadi-Needham (BAN) logic, and the Real-or-Random (ROR) model. A detailed comparative study on existing competing schemes including the analyzed scheme demonstrates that the devised framework furnishes better security features while also having lower computation costs and comparable communication costs than the existing schemes.

INDEX TERMS Digital twin, blockchain, authentication, key agreement, security.

I. INTRODUCTION

A Digital Twin is a real-time digital replica of a physical system that accurately reflects its features. The DT environment involves the formation of a clone of the tangible object to perform simulations in the virtual space. Grieves and Vickers [1] first proposed the idea of performing simulations with a clone in a virtual environment in 2002, and National Aeronautics and Space Administration (NASA) in 2010 referred to the method as a DT [2]. The DT concept was developed to make it possible to reap the benefits of paradigms like Industry 4.0 and the industrial Internet of Things (IIoT). The idea is to make every product or process-related data source and control interface description accessible through a single interface for

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

automatic communication establishment and auto-discovery. Without specific knowledge of each component, developers and engineers can determine, design, and construct the required interfaces, integrations, and communication links by analyzing the DTs of the incorporated components [3]. The devices may eventually be able to locate and communicate with one another without the need for a human engineer to stand in between them. With the assistance of DTs, this kind of auto-discovery and auto-established communication may eventually make IoT more scalable for currently unimaginable applications. The numerous fields in which DT technology is being studied are manufacturing, construction, healthcare, and space industries. IoT and mobile devices have recently been added to the DT technology's application range. For instance, autonomous driving can be achieved in a vehicular environment, and precise and detailed remote medical treatment can be carried out in a medical environment.

Cloud computing is the most feasible approach for implementing DT services since it has prodigious advantages. It provides on-demand services, computing resources, ubiquitous network access, etc., making it suitable for the next-generation information technology architecture. In cloud-assisted DT environments, the data owners generate data from physical assets and disseminate it to the cloud server, simulating DT in virtual space and sharing the simulation results with the owner. At the same time, the user can access the data upon request. However, putting DT technology into practice faces several obstacles. The biggest challenge is finding a secure way to share simulation and real-time data. Serious privacy implications are to be faced if the sensitive information transmitted by the data owner gets held by the adversary. Evidently, the below-illustrated points are necessary for the deployment of DT environment: (a) There is a strong urge to develop a secure medium for efficiently sharing the transmitted data. (b) There must be a procedure for validating the transmitted data; that is, verification of data integrity is required. (c) Security prerequisites such as untraceability, anonymity, and confidentiality should be guaranteed.

To achieve the aforementioned security prerequisites, we need a secure and privacy-preserving authentication protocol employing the benefits of blockchain technology. With blockchain, the data owner or user who utilizes data is allowed to verify the integrity of the data [4], [5], [6]. Users may readily validate the requested data using a Merkle hash tree. The framework proposed in this paper utilizes a cloud server to store the DT data and blockchain for the data hash values, enabling the users to verify the integrity of received data. Furthermore, the log transactions of shared data among the user-server are uploaded to the blockchain.

A. MOTIVATION AND CONTRIBUTIONS

Several authentication mechanisms [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] are introduced in the literature however, the majority of them cannot withstand various security assaults. For instance, many two-factor-based protocols cannot facilitate forward secrecy and user anonymity properties; many cannot withstand identity and password-guessing attacks. Similarly, some cannot withstand user and server impersonation attacks, and only a small number can be validated using ROR Model and BAN logic. Furthermore, most authentication mechanisms are designed employing traditional public cryptosystems and identity-based cryptosystems. However, these cryptosystems have some loopholes. The loopholes in the paradigms created using the public cryptosystem and the identity-based cryptosystem are the complex certificate management, storage, and key escrow problem, respectively. Since certificateless cryptosystems offer the best solution to the aforementioned issues, many certificateless paradigms have been proposed to overcome these vulnerabilities. In this paradigm, a third party is accountable for reckoning the partial private keys of users, while the user itself reckons the private key by employing the partial private key. Utilization of elliptic curve cryptography (ECC) in the system upsurges the computational efficiency. Hence, we have adopted the certificateless authentication scheme for the DT environment utilizing blockchain technology.

We could summarize our contributions as follows:

- Firstly, we review and cryptanalysis the scheme proposed by Son et al. [7] and identify that the scheme is susceptible to impersonation attacks, password guessing attacks, anonymity, and untraceability attacks. Besides, it does not support mutual authentication and session key agreement.
- We design a "secure three-factor privacy-preserving authentication scheme for the DT environment" by utilizing blockchain technology and "elliptic curve cryptography (ECC)" to realize secure communication among legitimate users and conquer security flaws.
- The suggested framework's informal analysis ensures that the protocol is resilient to various security assaults. Using the ROR model [17] and BAN logic [18], we also demonstrate that the proposed scheme can assure "mutual authentication" and "session key security".
- The computational and communication efficiency of the work is demonstrated by analyzing the presented work with the pre-existing authentication schemes.

B. STRUCTURE OF THE PAPER

The remaining structure of the paper is arranged as follows. Section II presents the related work. Section III is preliminaries which includes the threat model, bio-hashing function, and the security model. Section IV includes the review of Son et al.'s scheme [7], while Section V discusses the cryptanalysis of Son et al.'s scheme [7]. Section VI contains the proposed scheme to guarantee secure communication, whereas the security analysis of the proposed work containing informal analysis, BAN Logic and ROR model is given in Section VII. Section VIII includes a detailed comparative study of the proposed work with the existing competing schemes, and in the last, Section IX we have concluded our work.

II. RELATED WORK

In recent years, "access control and authentication" are widely-used two main security mechanisms in providing security in IoT-enabled environments [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30].

In 2002, Grieves [1] authoritatively introduced the concept and model of the Digital Twin as the applied paradigm underlying Product Lifecycle Management (PLM). Since the 1960, NASA has been refining the concept, which received recognition in 2010 when it was named digital twin [2]. We begin by outlining a few studies that can help explain the DT environment. A DT reference architecture is proposed by Aheleroff et al. [31] for industrial applications. They concentrated on establishing the Industry 4.0 DT reference architecture paradigm and included a DT as a server.

A secure and privacy-preserving protocol for DT-based traffic control is proposed by Lai et al. [32]. To enable data source authentication with efficient member revocation and privacy protection throughout the data uploading phase, the protocol adopts a group signature with a time-bound keys approach. After synchronization with its twin, this guarantees that data can be safely kept on cloud service providers. To enable flexibility and effective data sharing, an additional attribute-based access control approach is implemented in the data sharing phase. A cloud-based paradigm for healthcare services utilizing DT technology is proposed by Liu et al. [33]. Their main goal is integrating healthcare for elderly patients with digital twin technologies. According to their protocol, medical gadgets like radio frequency identification (RFID) cards, portable electrocardiograms, and wristbands generate health data, which is then gathered on computers or cell phones. The acquired data are subsequently transmitted across wireless networks, including mobile networks, Ethernet, and Wi-Fi, to a distant cloud server. A DT is used by Liu et al. [33] to build a conceptual model for cloud-based healthcare systems.

Further, there have been numerous attempts to integrate blockchain and DT technology. As per the management needs of 6G DTs-driven Internet of vehicles (IoV), a blockchain-based secure communication architecture has been designed by Liu et al. [34]. These systems can spot possible vehicle node threats while gaining access to data. Utilizing the blockchain will increase the precision and effectiveness of access control. A blockchain-based data management system for digital twins of products was presented by Huang et al. [35]. The blockchain is employed to efficiently and securely share, store, access, and authenticate digital twin data. For Industrial Internet of Things (IIoT) applications, the protocol designed by Sasikumar et al. [36] integrates DT with a distributed network employing blockchain. In order to deliver high-quality services for the IIoT, such as data privacy and security, this study suggests a Proof of Authority (PoA) trust mechanism based on blockchain technology. Similar to this, Wang et al. [37] suggested a sustainable DT management architecture for an IoT environment utilizing blockchain to enable network decentralization and efficient data transmission.

Grover et al. [11] highlighted the security vulnerabilities of the protocol designed by Wazid et al. [12]. Also, they proposed an enhanced mechanism for smart grid environments, which was analyzed by using the ProVerif tool. Kaur and Kumar [13] devised a two-factor user authentication framework for smart homes. They illustrated their scheme to be more efficient and highlighted the security vulnerabilities of the Shuai et al. [14] scheme.

Similarly, the security flaws of Chen et al. [15] work is illuminated by Wu et al. [8]. They further proved the superiority of their protocol by comparing it with similar pre-existing protocols. In telecare medical information systems(TMIS), Khatoon et al. [9] established a key agreement mechanism between clients and servers. They showed that their protocol could ensure several security functionalities with better efficiency. However, their scheme offers no mechanism for data verification and is susceptible to Known session-specific temporary information attack [16].

Sengupta et al. [10] also developed an authentication framework for cyber-physical systems utilizing ECC and bilinear pairing. However, this framework was later crypt-analyzed to show that it did not successfully preserve user anonymity. All these aforementioned procedures were developed for environments comparable to DT, but do not handle DT environments.

III. PRELIMINARIES

In this section, we discuss some preliminaries related to the threat model, bio-hashing function, and system model that are needed to discuss and analyze the proposed scheme in this paper.

A. THREAT MODEL

To demonstrate the security of the proposed scheme, the well-known Dolev-Yao (DY) model [38], [39] is presented in this section. The following are the capabilities of a malicious adversary in the DY model:

- A malicious adversary can replay, insert, eavesdrop, modify and delete transmitted messages sent through an open channel.
- An adversary can use the "power-analysis attacks to extract the secret credentials stored on a stolen user's smart card or mobile device".
- During the registration phase, the adversary can capture or tamper smart device. As a result, an adversary is able to obtain the secret credentials from the device's memory and can attempt various other security attacks.

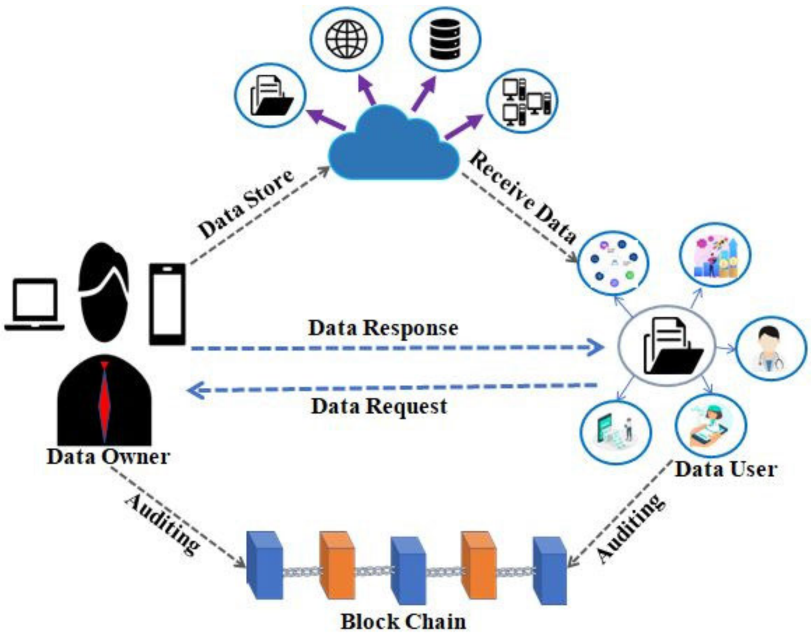

FIGURE 1. System model.

- Adversary could be a registered user or a malicious insider or vice-versa.
- Adversary can simultaneously perform offline identity and password guessing attacks. As a result, the adversary is able to simultaneously determine the genuine user's identity and password.

B. BIO-HASHING FUNCTION

A suitable method for identifying the authenticity of a user is the usage of the user's biometric information as an additional factor in an authentication system. Reference [40] demonstrated that fingerprint data of users could be converted to a bit form using biohashing and introduced a biohashing function that uses fingerprint data to verify users.

- A vector $u \in R^n$ is used to represent the biometric feature that is extracted from the fingerprint.
- Blum-Blum-Shub method is employed to generate a set of random nonces $s_i \in R^n$ $(i = 1, 2, \cdots, n)$.
- The basis s_i can be transformed into an orthonormal set of matrices $s_i \in R^n$ $(i = 1, 2, \ldots, n)$ using the Gram-Schmidt process.
- Compute the inner product between $s_i \in R^n(i = 1, 2, \cdots, n)$ and u, the resultant biohash code b_i is computed as

$$\mathrm{b_i} = \begin{cases} 0, & \text{if } \langle \mathrm{u} \mid \mathrm{r_i} \rangle \leq \tau \\ 1, & \text{if } \langle \mathrm{u} \mid \mathrm{r_i} \rangle > \tau \end{cases}$$

where τ denotes preset threshold.

C. SYSTEM MODEL

The blockchain-based system model for cloud-based digital twin environments is discussed in this section. There are five distinct entities in the proposed system model: trace authority, a cloud server, a data owner, a data user, and a blockchain as shown in Fig. 1. The following are the in-depth descriptions of each entity:

- Trace authority (TA): This entity is the trusted third party that is accountable for the generation of the system parameters and the partial private keys for the entities along with the participant's registration.
- Cloud Server (S): After a mutual key agreement, the cloud server receives data from the data owner, simulates DT in virtual space, and shares the simulation results with the owner. Additionally, the server can share DT data with the user after the user-owner mutual authentication. It also uploads hash values of log and stored data to the blockchain.
- Data Owner (O): This participant is responsible for collecting data from physical assets such as a wristband or a sensor. Once mutual authentication between both entities holds, the generated data is transmitted to the cloud server. In addition, giving access to the server to share data with a data user occurs when a data owner receives a request for data from a data user. The blockchain enables the data owner to examine the log record of the shared data.
- Data User (U): As per the requirement of data, the data user's request for DT data. Once the mutual authentication between the owner-user holds, the user can access the DT data stored over the cloud. The verification of data can be done utilizing blockchain technology.
- Blockchain: Blockchain stores log records between the data users and cloud server as well as the hash values of the data that is stored on the server. These log records help users to ensure whether the data is shared with the

TABLE 1. Notations of the devised framework.

Symbol	Description
O_s	s-th data owner
ID_s, PW_s	Identity, Password of O_s
SID_s	Secret-identity of O_s
HID_s	Pseudo-identity of O_s
S_i	i-th cloud server
U_r	r-th data user
u_r, u_s	Random nonce
b_r, b_s	Secret key of U_r, O_s
L_r, L_s	Message digest of U_r, O_s
Req	Request message of U_r
R_{is}, R_{si}	Exchanged Diffie-Hellman key between U_r and O_s
SK	Session key
$\oplus$	Bitwise exclusive-OR (XOR) operation
$\|\|$	Concatenation operation
$\cdot$	Multiplication operation
T_1, T_2, T_3, T_4	Time stamps
$A \dashrightarrow B$: Msg	Entity A sends the message, Msg, to entity B via secure channel
$A \rightarrow B$: Msg	Entity A sends the message, Msg, to entity B via public channel

authorized user or not. In addition, data users use the data hash values to ensure that the data have not been altered. After the smart contract has verified the signature of each transaction, it is uploaded.

IV. REVIEW OF SON ET AL.'s SCHEME

This section reviews the scheme proposed by Son et al. [7]. The notations used are mentioned in Table 1.

A. INITIALIZATION PHASE

In the initialization phase, TA selects a non-singular elliptic curve $E_q(j,k)$: $x^2 = y^3 + jy + k(\mod q)$ and two constants $j, k \in Z_q$ such that $4j^3 + 27k^2 \neq 0 \pmod q$, where q denotes a large prime number and reckons the partial private keys for all the entities involved. Then TA selects a base point P on $E_q(j,k)$, a secret key K_{TA}, and computes $P_{TA} = K_{TA}.P$. Afterwards, TA selects two multiplicative groups G and G_t such that $e : G \times G \rightarrow G_t$. Then TA selects two "cryptographic hash functions" defined as $h(.) : \{0,1\}^* \rightarrow Z_q, H(.) : \{0,1\}^* \rightarrow G$ and the system parameters $\{G_t, G, P_{TA}, P, h(.), h_b(.), H(.)\}$ are published.

B. REGISTRATION PHASE

During the registration phase, each entity involved in the protocol has to get registered with TA to participate in the network. Firstly TA selects ID_i and r_i for S_i and computes $P_i = r_i.P$, where the former denotes the private key while the latter denotes the public key of S_i. Further, O_s will register with the TA by utilizing its smart device D_s.

1) O_s selects ID_s, PW_s and selects a random nonce $g_s \in Z_q$. Then O_s computes $HID_s = H(ID_s||PW_s||g_s)$. Afterwards, $O_s \dashrightarrow TA : \{ID_s, HID_s\}$.
2) After receiving the message, TA will verify the freshness of ID_s to avoid re-registration. If not fresh, the process will be terminated. Otherwise, TA generates $r_s, n \in \{2^5, 2^{10}\}$, where n denotes the fuzzy verifier and computes $SID_s = r_s.HID_s$, $P_s = r_s.P$. Afterwards $TA \dashrightarrow O_s : \{SID_s, r_s, n\}$.
3) After receiving the message, O_s computes $HPW_s = h(ID_s||PW_s)$, $A_s = g_s \oplus HPW_s$ and $C_s = r_s \oplus h(g_s||HPW_s)$, $E_s = SID_s \oplus h(r_s||g_s||HPW_s)$ and $Auth_s = h(r_s||g_s||SID_s)(\mod n)$. Finally O_s stores $\{A_s, C_s, E_s, Auth_s, n\}$ in D_s.

C. AUTHENTICATION PHASE OF CLOUD-OWNER

Firstly O_s authenticates S_i to transmit the data initiated with their physical assets.

1) O_s inputs ID_s, PW_s into D_s, then D_s computes $HPW_s = h(ID_s||PW_s)$, $g_s = A_s \oplus HPW_s$, $r_s = C_s \oplus h(g_s||HPW_s)$, $SID_s = E_s \oplus h(r_s||g_s||HPW_s)$ and checks $Auth_s \stackrel{?}{=} h(r_s||g_s||SID_s)(\mod n)$. If the verification holds, D_s generates c_s, T_1 and therefore computes $HID_s = H(ID_s||PW_s||g_s)$ $R_s = c_s.g_s.P$, $R_{si} = c_s.g_s.P_i$, $PID_s = HID_s \oplus h(R_{si}||T_1)$ and $X_s = SID_s.h(HID_s||R_{si}||T_1)$. Thus $O_s \rightarrow S_i : \{R_s, PID_s, X_s, T_1\}$.
2) After receiving the message S_i first verifies $|T_1 - T_1^*| < \Delta T$. Then S_i computes $R_{si} = r_i.R_s$, $HID_s = PID_s \oplus h(R_{si}||T_1)$ and checks $\check{e}(X_s, P) \stackrel{?}{=} \check{e}(HID_s.h(HID_s||R_{si}||T_1), P_{TA})$. If the verification holds, S_i generates $c_i \in Z_q^*$, T_2 and therefore computes $R_i = c_i.P$, $R_{is} = c_i.R_s$. Afterwards S_i computes $SK_{is} = h(R_{si}||R_{is}||HID_s)$, $L_i \stackrel{?}{=} h(SK_{is}||R_{si}||R_{is}||T_2)$ and $S_i \rightarrow O_s : \{R_i, L_i, T_2\}$.
3) After receiving the message, O_s first verifies $|T_2 - T_2^*| < \Delta T$ and then computes $R_{is} = c_s.g_s.R_i$, $SK_{si} = h(R_{si}||R_{is}||HID_s)$. Afterwards O_s checks $L_i \stackrel{?}{=} h(SK_{is}||R_{si}||R_{is}||T_2)$.

D. AUTHENTICATION PHASE OF USER-OWNER

1) U_r generates a request message $Req_r \in Z_q$, selects a random nonce $u_r \in Z_q$ and timestamp T_3. Then U_r computes $Q_r = u_r.g_r.P$, where g_r denotes the random nonce selected in the registration phase of the user, and $U_{rs} = u_r.g_r.P_s$. Afterwards U_r computes $PID_r = HID_r \oplus h(U_{rs}||T_3)$, $M_r = Req_r \oplus h(HID_r||U_{rs}||T_3)$ and $X_r = SID_r.h(HID_r||Req_r||U_{rs}||T_3)$. Then $U_r \rightarrow O_s : \{Q_r, PID_r, M_r, X_r, T_3\}$.
2) Once the message has been received, O_s verifies $|T_3 - T_3^*| < \Delta T$ and computes $U_{rs} = r_r.Q_r$, $HID_r = PID_r \oplus h(U_{rs}||T_3)$, $Req_r = M_r \oplus h(HID_r||U_{rs}||T_3)$, and checks $\check{e}(X_r, P) \stackrel{?}{=} \check{e}(HID_r.h(HID_r||Req_r||U_{rs}||T_3), P_{TA})$. If the verification holds, O_s generates a random nonce $u_s \in Z_p^*$ and timestamp T_4. Then O_s computes $U_s = u_s.P$, $U_{sr} = u_s.X_r$, $SK_{sr} = h(U_{rs}||U_{sr}||HID_r||HID_s)$, and $L_s \stackrel{?}{=} h(SK_{sr}||U_{rs}||U_{sr}||T_4)$. Afterwards $O_s \rightarrow U_r : \{U_s, L_s, T_4\}$.
3) Once the message has been received, U_r first checks $|T_4 - T_4^*| < \Delta T$ and then further computes $U_{sr} = u_s.X_r$,

$SK_{sr} = h(U_{rs}||U_{sr}||HID_r||HID_s)$, and $L_s \stackrel{?}{=} h(SK_{sr}||U_{rs}||U_{sr}||T_4)$.

V. CRYPTANALYSIS OF SON ET AL.'s SCHEME

In this section, we present the security analysis of Son et al.'s framework [7].

A. OFFLINE PASSWORD GUESSING ATTACK

Assume that E is the privileged insider that belongs to TA. Therefore, the secret values ID_s, HID_s are known to E. Also, if E steals the owner's smart device D_s, he can obtain the parameters stored in it by using a side-channel analysis attack. Then with the assistance of a Privileged insider and smart card stolen attack, he can guess the password in the following manner:

Suppose E guesses the PW_s^* by utilizing the dictionary space and computes $HPW_s^* = h(ID_s||PW_s^*)$. Further E computes $a_s^* = A_s \oplus HPW_s^*$ and $HID_s^* = H(ID_s||PW_s^*||a_s^*)$. If $HID_s^* \stackrel{?}{=} HID_s$ holds, then the offline password-guessing attack is feasible. Therefore, the proposed framework is vulnerable to "offline password-guessing attacks".

B. IMPERSONATION ATTACKS

Firstly, the impersonation attacks are applied over the authentication phase between O_s and S_i.

1) OWNER IMPERSONATION ATTACK

In this attack, E obstructs the login message $\{R_s, PID_s, X_s, T_1\}$ sent by O_s through the public channel and uses side-channel attacks to extract all parameters from D_s. This attack demonstrates how E tries to impersonate the legitimate owner of Son et al.'s scheme. E generates $r_s^* \in Z_p^*$ and T_1^*. By using the above-mentioned Privileged insider attack, stolen device attack, the value HID_s is known to E, and therefore he can compute $r_s = C_s \oplus h(g_s||HPW_s)$, $SID_s = D_s \oplus h(r_s||g_s||HPW_s)$. Further E computes $R_s^* = r_s^*.g_s.P$, $R_{si}^* = r_s^*.g_s.P_i$, $PID_s^* = HID_s \oplus h(R_{si}^*||T_1^*)$ and $X_s^* = SID_s.h(HID_s||R_{si}^*||T_1^*)$ and $E \rightarrow S_i : \{R_s^*, PID_s^*, X_s^*, T_1^*\}$. This login message will survive the authentication test as it contains the valid ID_s, PW_s, g_s in addition to a fresh time stamp T_1^*.

2) CLOUD-SERVER IMPERSONATION ATTACK

This attack demonstrates how E tries to impersonate the legitimate server of Son et al. scheme [7] by obstructing the response message $\{R_i, L_i, T_2\}$. If TA gets malicious, then r_i can be obtained. Thus E generates $c_i^* \in Z_q^*$, T_2^* and computes $R_i^* = c_i^*.P$, $R_{si} = R_s.r_i$, $R_{is}^* = c_i^*.R_s$. Afterwards S_i computes $SK_i^* = h(R_{si}||R_{is}^*||HID_s)$, $L_i^* \stackrel{?}{=} h(SK_i||R_{si}||R_{is}^*||T_2^*)$ and $E \rightarrow O_s : \{R_i^*, L_i^*, T_2^*\}$. This response message will survive the authentication test as it contains the valid ID_s,PW_s, g_s, R_{si}, HID_s in addition to a fresh time stamp T_2^*. Here, the impersonation attack is applied over the authentication phase between U_r and O_s.

3) DATA USER IMPERSONATION ATTACK

This attack demonstrates how E tries to impersonate the legitimate user of Son et al. scheme [7] by obstructing the login message $\{Q_r, PID_r, M_r, X_r, T_3\}$. Firstly E generates a request message of its own $Req_r^* \in Z_p^*$, selects a random nonce u_r^* and timestamp T_3^*. Then E computes $U_r^* = u_r^*.g_r.P$, where g_r denotes the random nonce selected in the registration phase of the user, and $U_{rs}^* = u_r^*.g_r.P_s$. Afterwards E computes $PID_r^* = HID_r \oplus h(U_{rs}^*||T_3^*)$, $M_r^* = Req_r^* \oplus h(HID_r||U_{rs}^*||T_3^*)$ and $X_r^* = SID_r^*.h(HID_r||Req_r^*||U_{rs}||T_3^*)$. Then $E \rightarrow O_s : \{U_r^*, PID_r^*, M_r^*, X_r^*, T_3^*\}$. This login message will survive the authentication test as it contains the valid ID_s,PW_s, g_s, HID_r in addition to a fresh time stamp T_3^*.

4) OWNER IMPERSONATION ATTACK

In this attack, E obstructs the response message $\{U_s, L_s, T_4\}$ sent by O_s through the public channel and uses side-channel attacks to extract all parameters from D_s. Here E impersonates a legitimate owner of Son et al. scheme [7] and authenticates with another entity in the following manner. E generates a random nonce $u_s^* \in Z_p^*$ and timestamp T_4^*. From above mentioned privileged insider attack, r_s can be obtained. Then E computes $U_s^* = u_s^*.P$,$U_{rs} = Q_r.r_s$, $U_{sr}^* = u_s^*.X_r$ and $HID_r = PID_r \oplus h(U_{rs}||T_3)$. In the similar manner, E can compute $HID_s = h(ID_s||PW_s||g_s)$. Further E computes the session key $SK_{sr}^* = h(U_{rs}||U_{sr}^*||HID_r||HID_s)$, and $L_s^* \stackrel{?}{=} h(SK_{sr}^*||U_{rs}||U_{sr}^*||T_4^*)$. Afterwards $E \rightarrow U_r : \{U_s^*, L_s^*, T_4^*\}$. This response message will survive the authentication test as it contains the valid ID_s,PW_s, g_s, U_{rs}, HID_r, HID_s in addition to a fresh time stamp T_4^*. Therefore, the proposed protocol is vulnerable to all types of impersonation attacks.

C. KNOWN SESSION-SPECIFIC TEMPORARY INFORMATION ATTACK (KSSTIA)

In this attack, it is assumed that the session random nonce is leaked. Further, we have to compute the session key i.e.; it is believed that u_r and u_s are known to E, in addition to Q_r, PID_r, X_r and T_3. In order to compute the session key SK_{sr}, firstly the parameters U_{sr}, U_{rs}, HID_r, and HID_s must be known to E. Therefore, E computes $U_{sr} = u_r.X_r$, $HID_s = H(ID_s||PW_s||g_s)$. By above mentioned privileged insider attack, r_s is known to E and therefore he can compute $U_{rs} = Q_r.r_s$, $HID_r = PID_r \oplus h(U_{rs}||T_3)$. i.e, $SK_{sr} = h(U_{rs}||U_{sr}||HID_r||HID_s)$ can be computed. Thus the proposed protocol is vulnerable to KSSTIA.

D. ANONYMITY AND UNTRACEABILITY ATTACK

In this attack, E tries to trace O_s or U_r by utilizing the messages transmitted through the unsecured channels. Moreover, by using the above-mentioned Privileged insider attack, the pseudo identities HID_s and HID_r of O_s and U_r are disclosed to E. Thus he can easily trace O_s or U_r. Therefore the proposed protocol is vulnerable to anonymity and untraceability attack.

Owner O_s	Server S_i
Inputs ID_s, PW_s and B_s Computes $a_s = h_b(B_s)$ $HID_s = H(ID_s\|\|PW_s\|\|a_s)$ Sends $\{ID_s, HID_s\}$	
	TA verifies freshness of ID_s. Generates $r_s, n \in \{2^5, 2^{10}\}$, where n is the fuzzy verifier Computes $SID_s = r_s.HID_s$ Sends $\{SID_s, r_s, n\}$
Computes $A_s = r_s \oplus h(a_s\|\|HID_s)$ $C_s = SID_s \oplus h(r_s\|\|a_s\|\|HID_s)$ $Auth_s = h(r_s\|\|a_s\|\|SID_s)(\mod n)$ Stores $\{A_s, C_s, Auth_s, h_b(.), h(.),$ $H(.)\}$ in D_s	

FIGURE 2. Registration phase of O_s.

E. NO MUTUAL AUTHENTICATION

Son et al. [7] claimed that their protocol supports mutual authentication. However, we have found that authentication does not hold. Once the O_s receives the data request message from U_r, he first verifies the time stamp condition $|T_3 - T_3^*| \leq \triangle T$. Then he computes the key U_{rs} as $U_{rs} = r_r.Q_r$. Since the computation of U_{rs}, involves the user's private key r_r, which is generated by TA, O_s has no access to the user's private key. Thus, this depicts the design flaw of the designed protocol.

F. NO SESSION KEY AGREEMENT

Son et al. [7] claimed that in the designed protocol, both the U_r and O_s shares a common session key. Once the U_r receives the response message from O_s, he verifies the time stamp condition $|T_4 - T_4^*| \leq \triangle T$. Then he computes the key U_{sr} as $U_{sr} = u_s.X_r$. Since the computation of U_{sr} involves random nonce u_s generated by O_s, therefore U_r cannot compute U_{sr}. Consequently, the SK_{sr} cannot be computed. Hence the proposed framework has no session key agreement.

VI. CLOUD-ASSISTED BLOCKCHAIN-ENABLED SECURE COMMUNICATION FRAMEWORK

To mitigate the mentioned attacks on the Son et al.'s scheme [7], we now discuss an effective and improved scheme below.

A. INITIALIZATION PHASE

In the initialization phase, *TA* selects a non-singular elliptic curve $E_q(j, k) : x^2 = y^3 + jy + k(modq)$, two constants $j, k \in Z_q$ such that $4j^3 + 27k^2 \neq 0(modq)$, where q denotes a large prime number. Then *TA* selects a base point P on $E_q(j, k)$, a secret key K_{TA}, and computes $P_{TA} = K_{TA}.P$. Afterwards, *TA* selects two "multiplicative groups G and G_t such that $e : G \times G \to G_t, h(.) : \{0, 1\}^* \to Z_q, H(.) : \{0, 1\}^* \to G$" and the system parameters $\{G_t, G, q, P_{TA}, P, h(.), h_b(.), H(.)\}$ are published. It is worth noticing that one can also utilize the widely-accepted "fuzzy extractor technique" for biometric verification which is applied in designing the other protocols [41], [42].

B. REGISTRATION PHASE

During the registration phase, each entity involved in the protocol, such as the owner, user, and cloud server, has to get registered with the trace authority, via a secure channel (for example, via in-person).

1) O_s selects a unique ID_s, PW_s and imprints biometric B_s. Then O_s computes $a_s = h_b(B_s)$ and $HID_s = H(ID_s||PW_s||a_s)$. Afterwards $O_s \dashrightarrow TA$: $\{ID_s, HID_s\}$.
2) After receiving the message, *TA* will verify the freshness of ID_s to avoid re-registration. If not fresh, the process will be terminated. Otherwise, *TA* generates $r_s, n \in \{2^5, 2^{10}\}$, where n denotes the fuzzy verifier and computes $SID_s = r_s.HID_s$. Afterwards, $TA \dashrightarrow O_s : \{SID_s, r_s, n\}$.
3) After receiving the message, O_s computes $A_s = r_s \oplus h(a_s||HID_s)$, $C_s = SID_S \oplus h(r_s||a_s||HID_s)$ and $Auth_s = h(r_s||a_s||SID_s)(\mod n)$. Finally O_s stores $\{A_s, C_s, Auth_s, h_b(.), h(.), H(.)\}$ in D_s.

The summary of this registration phase is given in Fig. 2.

C. AUTHENTICATION PHASE OF CLOUD-OWNER

In this phase, O_s authenticates S_i to transmit the data initiated with their physical assets. Firstly O_s selects b_s as his private key and computes $P_s = b_s.P$ as his public key. The following are the steps needed for this phase:

1) O_s inputs ID_s, PW_s, B_s into D_s, then D_s computes $a_s = h_b(B_s)$, $HID_s = H(ID_s||PW_s||a_s)$, $r_s = A_s \oplus h(a_s||HID_s)$, $SID_s = C_s \oplus h(r_s\,||a_s\,||HID_s)$ and checks $Auth_s \stackrel{?}{=} h(r_s\,||a_s\,||SID_s) \pmod n$. If the verification holds, D_s generates r_s, T_1 and therefore computes $R_s = r_s.b_s.P$, $R_{si} = r_s.b_s.P_i$, $PID_s = HID_s \oplus h(R_{si}\,||T_1)$ and $X_s = R_{si}.h(HID_s\,||SID_s\,||T_1)$. Thus, $O_s \to S_i : \{R_s, PID_s, X_s, T_1\}$.
2) After receiving the message S_i first verifies $|T_1 - T_1^*| \leq \triangle T$. Then S_i computes $R_{si} = b_i.R_s$, $HID_s = PID_s \oplus h(R_{si}\,||T_1)$ and checks $\check{e}(X_s, P) \stackrel{?}{=} \check{e}(R_s.h(HID_s\,||SID_s\,||T_1), P_i)$. If the verification holds, S_i generates r_i, T_2 and therefore computes $R_i = r_i.b_i.P$, $R_{is} = r_i.b_i.P_s$. Afterwards, S_i computes $SK_{is} = h(R_{si}\,||R_{is}\,||HID_s\,||T_2)$, $L_i \stackrel{?}{=} h(SK_{is}\,||R_{si}||\,R_{is}\,||T_2)$ and $S_i \to O_s : \{R_i, L_i, T_2\}$.
3) After receiving the message, O_s first checks $|T_2 - T_2^*| \leq \triangle T$ and then computes $R_{is} = b_s.R_i$, $SK_{si} = h(R_{si}\,||R_{is}\,||HID_s\,||T_2)$ and verifies $L_i \stackrel{?}{=} h(SK_{si}\,||R_{si}\,||R_{is}\,||T_2)$.

The summary of the login and authentication phase between O_s and S_i is provided in Fig. 3.

D. AUTHENTICATION PHASE OF USER-OWNER

This phase allows the U_r to request data from O_s. The following are the steps needed for this phase:

1) U_r generates a request message $Req_r \in Z_p$, selects a random nonce $u_r \in Z_p$ and timestamp T_3. Then U_r computes $Q_r = a_r.u_r.P$, where $a_r = h_b(B_r)$, and

Owner O_s		Server S_i
Inputs ID_s, PWs and B_s into D_s D_s computes $a_s = h_b(B_s)$ $HID_s = H(ID_s\|\|PW_s\|\|a_s)$ $r_s = A_s \oplus h(a_s\|\|HID_s)$ $SID_s = C_s \oplus h(r_s\|\|a_s\|\|HID_s)$ Checks $Auth_s \stackrel{?}{=} h(r_s\|\|a_s\|\|SID_s) \pmod n$ Generates $r_s \in Z_p^*$, and timestamp T_1 Computes $R_s = r_s.b_s.P$ $R_{si} = r_s.b_s.P_i$ $PID_s = HID_s \oplus h(R_{si}\|\|T_1)$ $X_s = R_{si}.h(HID_s\|\|SID_s\|\|T_1)$		
	$\xrightarrow{\langle R_s, PID_s, X_s, T_1\rangle}$	
		Verifies if $\|T_1 - T_1^*\| \le \triangle T$ Computes $R_{si} = b_i.R_s$ $HID_s = PID_s \oplus h(R_{si}\|\|T_1)$ $\check{e}(X_s, P) \stackrel{?}{=} \check{e}(R_s.h(HID_s\|\|SID_s\|\|T_1), P_i)$ Generates $r_i \in Z_p^*$, T_2 Computes $R_i = r_i.b_i.P$ $R_{is} = r_i.b_i.P_s$ $SK_{is} = h(R_{si}\|\|R_{is}\|\|HID_s\|\|T_2)$ $L_i \stackrel{?}{=} h(SK_{is}\|\|R_{si}\|\|R_{is}\|\|T_2)$
	$\xleftarrow{\langle R_i, L_i, T_2\rangle}$	
Checks if $\|T_2 - T_2^*\| \le \triangle T$ If yes, computes $R_{is} = b_s.R_i$ $SK_{si} = h(R_{si}\|\|R_{is}\|\|HID_s\|\|T_2)$ Verifies $L_i \stackrel{?}{=} h(SK_{si}\|\|R_{si}\|\|R_{is}\|\|T_2)$		

FIGURE 3. Login and authentication phase between O_s and S_i.

$U_{rs} = u_r.a_r.P_s$. Afterwards U_r computes $PID_r = HID_r \oplus h(U_{rs}||T_3)$, $M_r = Req_r \oplus h(HID_r||U_{rs}||T_3)$ and $X_r = U_{rs}.h(HID_r||Req_r||SID_r||T_3)$. Then $U_r \rightarrow O_s$: $\{Q_r, PID_r, M_r, X_r, T_3\}$.

2) Once the message has been received, O_s verifies $|T_3 - T_3^*| \le \Delta T$ and computes $U_{rs} = b_s.Q_r$, $HID_r = PID_r \oplus h(U_{rs}||T_3)$, $Req_r = M_r \oplus h(HID_r||U_{rs}||T_3)$, and checks $\check{e}(X_r, P) \stackrel{?}{=} \check{e}(Q_r.h(HID_r||Req_r||SID_r||T_3), P_s)$. If the verification holds, O_s generates a random nonce $u_s \in Z_p^*$ and timestamp T_4. Then O_s computes $U_s = u_s.P$, $U_{sr} = U_s.X_r$, $SK_{sr} = h(U_{rs}||U_{sr}||HID_r||HID_s||T_4)$, and $L_s \stackrel{?}{=} h(SK_{sr}||U_{rs}||U_{sr}||T_4)$. Afterwards $O_s \rightarrow U_r$: $\{U_s, L_s, T_4\}$.

3) Once the message has been received, U_r first checks $|T_4 - T_4^*| \le \Delta T$ and then further computes $U_{sr} = U_s.X_r$, $SK_{sr} = h(U_{rs}||U_{sr}||HID_r||HID_s||T_4)$, and $L_s \stackrel{?}{=} h(SK_{sr}||U_{rs}||U_{sr}||T_4)$.

Authentication phase between U_r and O_s is summarized in Fig. 4.

E. SECURE DATA AGGREGATION PHASE

During the authentication between O_s and S_i (see Section VI-C), after mutual authentication both O_s and S_i established a session key SK_{is} ($= SK_{si}$) for their secret communications. Similarly, after the mutual authentication between U_r and O_s (see Section VI-D), both U_r and O_s also established a session key SK_{sr} for secret communications. Thus, using the secret session key SK_{is} ($= SK_{si}$), O_s securely transmits the data to the authenticated S_i, with their physical assets. Moreover, using the secret session key SK_{sr}, U_r also securely requests the data from O_s. In this way, secure data collection (aggregation) takes place by the respective entities in the network.

F. BLOCKCHAIN IMPLEMENTATION PHASE

This phase is used to form the transactions, say TX_j, from the authenticated aggregated data in Section VI-E. Next, the formed transactions TX_j are used to form various blocks. Each block consists of a threshold number thr_n of transactions, say $TX_1, TX_2, \cdots, TX_{thr_n}$. In addition, each block contains two parts: a) block header and b) block payload. The block header contains the following fields:

- **Block Version:** It is a unique serial number to the block.
- **Previous Block Hash:** The hash value of the previous block in the blockchain.

User U_r		Owner O_s
Generates a request message $Req_r \in Z_p$, selects a random nonce $u_r \in Z_p$ and T_3 Computes $Q_r = a_r.u_r.P$, where $a_r = h_b(B_r)$ $U_{rs} = u_r.a_r.P_s$ $PID_r = HID_r \oplus h(U_{rs}\|\|T_3)$ $M_r = Req_r \oplus h(HID_r\|\|U_{rs}\|\|T_3)$ $X_r = U_{rs}.h(HID_r\|\|Req_r\|\|SID_r\|\|T_3)$		
	$\xrightarrow{\langle Q_r, PID_r, M_r, X_r, T_3 \rangle}$	
		Checks $\|T_3 - T_3^*\| \leq \triangle T$ Computes $U_{rs} = b_s.Q_r$ $HID_r = PID_r \oplus h(U_{rs}\|\|T_3)$ $Req_r = M_r \oplus h(HID_r\|\|U_{rs}\|\|T_3)$ $\check{e}(X_r, P) \stackrel{?}{=} \check{e}(Q_r.h(HID_r\|\|Req_r\|\|SID_r\|\|T_3), P_i)$ Generates $u_s \in Z_p^*$, T_4 Computes $U_s = u_s.P$, $U_{sr} = U_s.X_r$ $SK_{sr} = h(U_{rs}\|\|U_{sr}\|\|HID_r\|\|HID_s\|\|T_4)$ Checks $L_s \stackrel{?}{=} h(SK_{sr}\|\|U_{rs}\|\|U_{sr}\|\|T_4)$
	$\xleftarrow{\langle U_s, L_s, T_4 \rangle}$	
Checks $\|T_4 - T_4^*\| \leq \triangle T$ Computes $U_{sr} = U_s.X_r$ $SK_{sr} = h(U_{rs}\|\|U_{sr}\|\|HID_r\|\|HID_s\|\|T_4)$ $L_s \stackrel{?}{=} h(SK_{sr}\|\|U_{rs}\|\|U_{sr}\|\|T_4)$		

FIGURE 4. **Authentication phase between U_r and O_s.**

- **Merkle Tree Root:** The root value of the Merkle tree constructed from a set of thr_n transactions, TX_1, $TX_2, \cdots, TX_{thr_n}$, in the block.
- **Timestamp:** The timestamp value when the block was created.
- **Block Owner:** The owner of the block's transactions.

The block payload contains the thr_n transactions, namely $TX_1, TX_2, \cdots, TX_{thr_n}$. Apart from the block payload, the current block hash is calculated as the hash of the block header and block payload. The structure of a typical block is depicted in Fig. 5.

Now, once a block, say $Block_i$, is formed, it is sent to the blockchain network. The blockchain network consists of a set of peer nodes which actively involve in the mining process in order to provide their consensus for approving and adding that block in the blockchain. In this paper, we have used the "voting-based Practical Byzantine Fault Tolerance (PBFT) consensus algorithm" [43] for verifying and adding the block $Block_i$ into the blockchain. For details of this voting based consensus algorithm, please refer to the work in [30].

Block Header	
Block Version	BVer
Previous Block Hash	PBHash
Merkle Tree Root	MTR
Timestamp	TS
Block Owner	Owner
Block Payload	
Transaction #1	TX_1
Transaction #2	TX_2
⋮	⋮
Transaction $\#thr_n$	TX_{thr_n}
Current Block Hash	CBHash

FIGURE 5. **Formation of a block $Block_i$ on thr_n transactions.**

VII. SECURITY ANALYSIS

In this section, we provide a detailed security analysis via the informal (heuristic) security analysis and also the formal security analysis using both the BAN logic and the Real-Or-Random (ROR) oracle model.

A. INFORMAL ANALYSIS

This section presents the security analysis of the devised framework.

1) REPLAY ATTACK

In every step of the devised framework, both the O_s and U_r generates random nonce u_r, u_s, and fresh timestamps T_3, T_4. Thus even if E tries to resend an old encapsulated message directly, he will not succeed in executing a replay attack. Since the message contains both the counter-measures, as a result, the U_r can determine the nature of the assault. Thus the devised framework is resilient to replay attacks.

2) PRIVILEGED-INSIDER ATTACK

In the registration phase of the devised framework, O_s imprints his biometric while making a registration request. The request message contains ID_s, HID_s, where $HID_s = h(ID_s||PW_s||a_s)$. Additionally, even if E utilizes the data of the stolen smart device, he will not succeed in guessing O_s password because the computation of HID_s involves the biometric of a user, as mentioned in the improved scheme. Thus the devised framework withstands privileged-insider attacks.

3) STOLEN SMART DEVICE ATTACK

Assume that E obtains D_s and extracts all the stored parameters $\{B_s, C_s, Auth_s, h_b(.), h(.), H(.)\}$. However, all the parameters are protected with XOR and hash operations using ID_s, PW_s and B_s. Therefore E cannot acquire sensitive information about U_s. Thus, the devised framework is resilient to Stolen smart device attacks.

4) OFFLINE PASSWORD GUESSING ATTACK

Assume that E intercepts the transmitted messages $U_r \rightarrow O_s : \{Q_r, PID_r, M_r, X_r, T_3\}$ and $O_s \rightarrow U_r : \{U_s, L_s, T_4\}$ sent over an insecure channel. Additionally, E extracts the parameters $\{B_s, C_s, Auth_s, h_b(.), h(.), H(.)\}$ stored in D_s. Since the transmitted messages do not contain the PW_s, E attempts to guess the owner's password by utilizing the dictionary space. However, the involvement of the bio-hashing function makes it difficult for the adversary to compute PW_s. Further, if somehow E guesses PW_s of O_s, he cannot verify the guessed values because $Auth_s$ is protected by using the fuzzy verifier n. Thus, the devised framework is resilient to Offline password-guessing attacks.

5) SESSION KEY COMPUTATION ATTACK

In our scheme, the computation of the session key $SK = h(U_{rs}||U_{sr}||HID_r||HID_s||T_4))$ depends upon the parameters U_{rs}, U_{sr}, HID_r, and HID_s. Since these parameters are not transmitted through the messages over the insecure channel thus, the adversary need to compute these values. Further, the security of the keys U_{sr}, U_{rs} relies on the difficulty of solving the "elliptic curve discrete logarithm problem". Additionally, in each session, both the O_s and U_r generate fresh nonce's, which makes it difficult for the adversary to compute SK. Thus, the proposed framework is resilient to session key computation attacks.

6) PERFECT FORWARD SECRECY

Assume that E intercepts the transmitted messages $U_r \rightarrow O_s : \{Q_r, PID_r, M_r, X_r, T_3\}$ and $O_s \rightarrow U_r : \{U_s, L_s, T_4\}$ sent over an insecure channel. Additionally, E obtains the long-term keys b_s, b_r and intends to compute SK. However, the adversary will not succeed in computing SK without the information of nonces. Thus, the devised framework guarantees perfect forward secrecy.

7) IMPERSONATION ATTACK

If an adversary A attempts to impersonate a legal user or owner, he/she has to generate the login request message $\{Q_r, PID_r, M_r, X_r, T_3\}$ or response message $\{U_s, L_s, T_4\}$. However, the computation of all parameters in both messages involves the usage of computed key U_{rs}, U_{sr}, whose security relies on the difficulty of solving the elliptic curve discrete logarithm problem. Further, the computation of parameters includes the random nonce and the bio-hashing function. Thus, the proposed framework is robust against impersonation attacks.

8) KNOWN SESSION SPECIFIC TEMPORARY INFORMATION ATTACK (KSSTIA)

Assume that the random nonce used in each session are leaked, and the adversary attempts to compute SK, i.e., u_s, u_r are known to E. Sk is computed by using U_{rs}, U_{sr}, HID_r, HID_s and T_4. There are only two ways to compute U_{rs} i.e, $U_{rs} = u_r.a_r.P_s$ or $U_{rs} = b_s.Q_r$. The first one includes the biometric of user a_r along with the random nonce generated u_r, whereas the second involves usage of the owner's private key b_s. Both of these are unknown to E. Similarly, other parameters U_{sr}, HID_r, and HID_s cannot be calculated, resulting E cannot compute SK. Therefore, the proposed framework is secure against KSSTIA attack.

9) ANONYMITY AND UNTRACEABILITY

An adversary E can utilize messages sent over insecure channels to trace an individual. E cannot, however, determine who sent the message because the pseudo-identities HID_s and HID_r are not revealed in the transmitted messages. Therefore the proposed framework assures anonymity and untraceability.

10) MUTUAL AUTHENTICATION

During the authentication phase, a login request message $\{Q_r, PID_r, M_r, X_r, T_3\}$ is sent to O_s. O_s first verifies the timestamp condition $|T_3 - T_3^*| < \triangle T$? and then computes U_{rs}, HID_r, Req_r, verifies $\check{e}(X_r, P) \stackrel{?}{=} \check{e}(Q_r.h(HID_r||Req_r||SID_r||T_3), P_i)$. If the verification holds, O_s authenticates U_r and sends the response message $\{U_s, L_s, T_4\}$ to U_r. Afterwards U_r also computes some values and verifies $L_s \stackrel{?}{=} h(SK_{sr}||U_{rs}||U_{sr}||T_4)$. The similar procedure is followed between O_s, S_i. Thus mutual authentication holds.

11) DATA VERIFICATION

The proposed framework assures data verification by utilizing blockchain technology. Once U_r has received the requested data from S_i, U_r can check the integrity of data using the hash values stored in the blockchain. If the values are not same, U_r can infer that the data have been altered and is invalid.

B. BAN LOGIC ANALYSIS

BAN logic is frequently used to demonstrate a protocol's mutual authentication. We use BAN logic in this

TABLE 2. Notations of BAN logic.

Symbol	Description
S_1, S_2	Statements
R_1, R_2	Principals
$R_1 \mid\sim S_1$	R_1 once said S_1
$R_1 \mid\equiv S_1$	R_1 believes S_1
$R_1 \Rightarrow S_1$	R_1 controls S_1
$R_1 \lhd S_1$	R_1 receives S_1
$\#S_1$	S_1 is fresh
$(S_1)_k$	S_1 is encrypted with key k
$R_1 \overset{k}{\leftrightarrow} R_2$	R_1 and R_2 communicate with shared key k
SK	Session key

section to demonstrate that the proposed scheme ensures mutual authentication. In order to carry out the BAN logic proof, we also introduce logical postulates, idealized forms, assumptions and goals. The notations used in BAN logic are listed in Table 2.

1. Logical Postulates

- The Message Meaning Rule (MMR):

$$\frac{R_1|\equiv R_1 \overset{k}{\leftrightarrow} R_2, R_1 \lhd \{S_1\}_k}{R_1|\equiv R_2|\sim S_1}$$

- The Nonce verification rule (NVR):

$$\frac{R_1|\equiv \#(S_1), R_1|\equiv R_2|\sim S_1}{R_1|\equiv R_2|\equiv S_1}$$

- The Jurisdiction Rule (JR):

$$\frac{R_1|\equiv R_2|\Rightarrow S_1, R_1|\equiv R_2\equiv S_1}{R_1|\equiv S_1}$$

- The Belief Rule (BR):

$$\frac{R_1|\equiv (S_1, S_2)}{R_1|\equiv S_1}$$

- Freshness Rule (FR):

$$\frac{R_1|\equiv \#(S_1)}{R_1|\equiv \#(S_1, S_2)}$$

2. Goals

The following are the goals for demonstrating the correctness of our framework:

- $GOAL^1 : U_r|\equiv U_r \overset{SK}{\leftrightarrow} O_s$
- $GOAL^2 : U_r|\equiv O_s|\equiv U_r \overset{SK}{\leftrightarrow} O_s$
- $GOAL^3 : O_s|\equiv U_r \overset{SK}{\leftrightarrow} O_s$
- $GOAL^4 : O_s|\equiv U_r|\equiv U_r \overset{SK}{\leftrightarrow} O_s$

3. Assumptions

The following are the assumptions of our BAN Logic protocol:

- $ASSUMPTION^1 : O_s|\equiv (U_r \overset{U_{sr}}{\leftrightarrow} O_s)$
- $ASSUMPTION^2 : O_s|\equiv \#(T_3)$
- $ASSUMPTION^3 : U_r|\equiv (U_r \overset{U_{rs}}{\leftrightarrow} O_s)$
- $ASSUMPTION^4 : U_r|\equiv \#(T_4)$
- $ASSUMPTION^5 : O_s|\equiv U_r \Rightarrow (U_r \overset{SK}{\leftrightarrow} O_s)$
- $ASSUMPTION^6 : U_r|\equiv O_s \Rightarrow (U_r \overset{SK}{\leftrightarrow} O_s)$

4. Idealized Forms

The idealized form of login and authentication messages $\{Q_r, PID_r, M_r, X_r, T_3\}$ and $\{U_s, L_s, T_4\}$ of our scheme are as follows:

- $MESSAGE^1 : U_r \rightarrow O_s : (Q_r, HID_r, T_3)_{U_{rs}}$
- $MESSAGE^2 : O_s \rightarrow U_r : (U_s, HID_s, T_4)_{U_{sr}}$

5. Proof Using Ban Logic

To prove the stated goals, the BAN logic proof employs the aforementioned logical postulates, assumptions, and idealized forms.

- From MESSAGE 1, we have G_1.
$$G_1 : O_s \lhd (Q_r, HID_r, T_3)_{U_{rs}}$$
- G_2 is obtained from G_1 and $ASSUMPTION^3$ by applying MMR.
$$G_2 : O_s|\equiv U_r|\sim (Q_r, HID_r, T_3)$$
- G_3 is obtained from G_1 and $ASSUMPTION^2$ by applying FR.
$$G_3 : O_s|\equiv \#(Q_r, HID_r, T_3)$$
- Combining G_2, G_3 and further applying NVR yields G_4.
$$G_4 : O_s|\equiv U_r|\equiv (Q_r, HID_r, T_3)$$
- Applying BR on G_4 yields G_5.
$$G_5 : O_s|\equiv U_r|\equiv (HID_r)$$
- From MESSAGE 2, we have G_6.
$$G_6 : U_r \lhd (U_s, HID_s, T_4)_{U_{sr}}$$
- Now G_7 is obtained from G_6 and $ASSUMPTION^1$ by applying MMR.
$$G_7 : U_r|\equiv O_s|\sim (U_s, HID_s, T_4)$$
- G_8 is obtained from G_6 and $ASSUMPTION^4$ by applying FR.
$$G_8 : U_r|\equiv \#(U_s, HID_s, T_4)$$
- Combining G_7, G_8 and further applying NVR yields G_9.
$$G_9 : U_r|\equiv O_s|\equiv (U_s, HID_s, T_4)$$
- Applying BR on G_9 yields G_{10}.
$$G_{10} : U_r|\equiv O_s|\equiv (HID_s)$$
- Using G_4 and G_5, O_s can generate the session key $SK_{sr} = h(U_{rs}||U_{sr}||HID_r||HID_s||T_4)$ and G_{11} can be obtained.
$$G_{11} : O_s|\equiv U_r|\equiv U_r \overset{SK}{\leftrightarrow} O_s \textbf{ (Goal-4)}$$
- G_{12} is obtained by using G_{11} and $ASSUMPTION^5$ following JR.
$$G_{12} : O_s|\equiv U_r \overset{SK}{\leftrightarrow} O_s \textbf{ (Goal-3)}$$
- Using G_9 and G_{10}, U_r can generate the session key $SK_{rs} = h(U_{rs}||U_{sr}||HID_r||HID_s||T_4)$ and G_{13} can be obtained.
$$G_{13} : U_r|\equiv O_s|\equiv U_r \overset{SK}{\leftrightarrow} O_s \textbf{ (Goal-2)}$$
- G_{14} is obtained by using G_{13} and $ASSUMPTION^6$ following JR.
$$G_{14} : U_r|\equiv U_r \overset{SK}{\leftrightarrow} O_s \textbf{ (Goal-1)}$$

TABLE 3. Different queries and their descriptions.

Queries	Descriptions
Execute($R^{t1}_{U_r}, R^{t2}_{O_s}$)	The adversary can obstruct messages sent through the public channel between $R^{t1}_{U_r}$ and $R^{t2}_{O_s}$.
CorruptD$_s$ ($R^{t2}_{O_s}$)	The D_s of $R^{t2}_{O_s}$ can be obtained by the adversary to extract the stored information.
Reveal(R^t)	The adversary can obtain the current session key SK_{rs} by utilizing this query.
Send(R^t, message)	The request message is sent to other participants by the adversary, who then receives the response message.
Test(R^t)	In this query, a coin b is tossed. After executing the Test query (R^t), (R^t) obtains a random number when $b = 0$ and a session key SK_{rs} when $b = 1$; obtains a null () otherwise. We can guarantee that our scheme protects the session key if the adversary cannot differentiate between the random number and the session key.

C. FORMAL SECURITY PROOF USING ROR MODEL

The ROR model is frequently used to demonstrate the security of various authentication protocols [44], [45], [46]. This section examines the session key security of the proposed framework by using the ROR model. We define $R^{t_1}_{U_r}$ and $R^{t_2}_{O_s}$ as participants such as r^{th} user and s^{th} owner, where t_i represents the instance of the participants. An adversary can perform *Execute*, *Send*, *Test*, and *CorruptD$_s$ queries* to carry out a variety of security attacks under the ROR model. The queries are described in Table 3.

Theorem 1: $Adv_s(t)$ is defined as the probability of breaking the proposed work's session key security in polynomial time t. Therefore the derived result is as follows.

$$Adv_s(t) \leq \frac{Q^2_{hash}}{|Hash|} + \frac{Q_s}{2^{l-1}|d_p|} + 2Adv^{ECDHP}_s(t)$$

where $Q_s, Q_{hash}, |Hash|, d_p, l$ and $Adv^{ECDHP}_s(t)$ represent "the number of send queries, the number of hash queries, the range space of hash function, size of password dictionary, the number of bits of biometric information and advantage of an adversary to break the elliptic curve decisional Diffie-Hellman problem (ECDHP)."

Proof: We have divided the formal proof into a sequence of five games G_j, where j=1,2,3,4,5. We define Sc^{adv}_j as the probability of the adversary winning the G_j. Additionally $Prob_s[Sc^{adv}_j]$ denotes the advantage of Sc^{adv}_j. The specific steps of each game are listed below.

- G_1 : This game G_1 simulates the attack game under the real protocol running conditions. The adversary does not conduct a query and has no information. As a result, the adversary selects a random bit b. Our protocol ensures the semantic security for *SK* by guessing random bit *b*. Then,

$$Adv_s(t) = |2Prob_s[Sc^{adv}_1] - 1| \tag{1}$$

- G_2 : The game G_2 implements the eavesdropping attack of adversary. At first the adversary performs $Execute(R^{t1}_{U_r}, R^{t2}_{O_s})$ query and obstructs the transmitted messages $\{Q_r, PID_r, M_r, X_r, T_3\}$, $\{U_s, L_s, T_4\}$ followed by the $Test(R^t)$ query to ascertain whether the returned result is SK_{rs} or not. The computation of SK_{rs} requires the secret values U_{rs}, U_{sr} along with the computed values HID_r, HID_s and T_4. However, the adversary is unable to obtain these values. Therefore adversary's probability of winning the G_2 is similar to that of G_1. Hence,

$$Prob_s[Sc^{adv}_1] = Prob_s[Sc^{adv}_2] \tag{2}$$

- G_3 : In order to calculate Sk_{rs}, the adversary uses both *Hash* and *Send* queries. The adversary can also use the messages $\{Q_r, PID_r, M_r, X_r, T_3\}$, $\{U_s, L_s, T_4\}$. Since these messages are protected by random numbers u_r, u_s, and hash functions, thus, to compute Sk_{rs}, the adversary should find the hash collision. After that, using the birthday paradox, we arrive at the following conclusion:

$$|Prob_s[Sc^{adv}_3] - Prob_s[Sc^{adv}_2]| \leq \frac{Q^2_{hash}}{2|Hash|} \tag{3}$$

- G_4 : In game G_4, the adversary attempts to obtain Sk_{rs} by using $CorruptD_s(R^{t2}_{O_s})$ query. Using a power analysis attack, the adversary can extract the secret credentials $A_s, C_s, Auth_s, n$ from the SC memory in G_4, where $A_s = r_s \oplus h(a_s||HID_s)$, $C_s = SID_s \oplus h(r_s||a_s||HID_s)$. The computation of Sk_{rs} requires the information of ID_s, PW_s, and B_s along with the random numbers. Consequently, using the password dictionary and biometric information of n bits, the adversary can attempt to guess values used to compute Sk_{rs}. Therefore, we then arrive at the following conclusion:

$$|Prob_s[Sc^{adv}_4] - Prob_s[Sc^{adv}_3]| \leq \frac{Q_s}{2^l|d_p|} \tag{4}$$

- G_5 : The adversary can also compute $SK_{sr} = h(U_{rs}||U_{sr}||HID_r||HID_s||T_4)$ by utilizing $\{Q_r, PID_r, M_r, X_r, T_3\}$, $\{U_s, L_s, T_4\}$. These messages contain Q_r, U_s, so the adversary can use it. Still, they cannot compute U_{rs}, U_{sr} as the security of both parameters relies on ECDHP. Therefore, we arrive at the following conclusion:

$$|Prob_s[Sc^{adv}_5] - Prob_s[Sc^{adv}_4]| \leq Adv^{ECDHP}_s(t). \tag{5}$$

Using $Test(R^t)$ query, the adversary tries to figure out the right bit b to win the game. As a result, we get the following outcome:

$$Prob_s[Sc^{adv}_5] = \frac{1}{2} \tag{6}$$

Thus combining the equations (1),(2) and (6) we get

$$\frac{1}{2}Adv_s(t) = |Prob_s[Sc^{adv}_1] - \frac{1}{2}| \tag{7}$$

$$= |Prob_s[Sc_2^{adv}] - \frac{1}{2}|$$
$$= |Prob_s[Sc_2^{adv}] - Prob_s[Sc_5^{adv}]|$$

Further, using triangular inequality, equations (3),(4),(5), and (7) can be transformed into the following:

$$\begin{aligned} &|Prob_s[Sc_2^{adv}] - Prob_s[Sc_5^{adv}]| \\ &\le |Prob_s[Sc_2^{adv}] - Prob_s[Sc_4^{adv}]| \\ &\quad + |Prob_s[Sc_4^{adv}] - Prob_s[Sc_5^{adv}]| \\ &\le |Prob_s[Sc_2^{adv}] - Prob_s[Sc_3^{adv}]| \\ &\quad + |Prob_s[Sc_3^{adv}] - Prob_s[Sc_4^{adv}]| \\ &\quad + |Prob_s[Sc_4^{adv}] - Prob_s[Sc_5^{adv}]| \\ &\le \frac{Q_{hash}^2}{2|Hash|} + \frac{Q_s}{2^l|d_p|} + Adv_s^{ECDHP}(t). \end{aligned} \tag{8}$$

Therefore, by combining (7)and (8), we obtain

$$Adv_s(t) \le \frac{Q_{hash}^2}{|Hash|} + \frac{Q_s}{2^{l-1}|d_p|} + 2Adv_s^{ECDHP}(t).$$

□

VIII. PERFORMANCE ANALYSIS

This section analyzes and compares the communication costs, computation costs, and security features of the proposed authentication protocol with other existing protocols in similar environments [7], [8], [9], [10].

A. SECURITY FEATURES

This section compares the proposed scheme's security features to those of previous schemes [7], [8], [9], [10]. Table 4 demonstrates that the proposed scheme resists various security attacks, namely replay attacks, offline password guessing attacks, privileged insider attacks, impersonation attacks, KSSTIA, perfect forward secrecy, stolen smart device attack, session key computation attack, anonymity, and untraceability attack. Additionally, our scheme offers data verification and mutual authentication. Therefore, the proposed scheme has a wider range of security features and offers superior security than the other existing schemes [7], [8], [9], [10].

B. COMPUTATIONAL COST

We refer to the Java pairing-based cryptography library-based experiments carried out in [8]. The experiment was carried out on a computer with 16 GB of memory and a 2.3 GHz Intel it-8300H quad-core processor. The time cost of each operation is described in Table 5. Following terms are used to compare the performances of different schemes.

Since a bitwise XOR operation, concatenation operation, and a one-way hash function takes very less computation time; therefore we neglect their cost during performance evaluation. We have considered the time cost for the login and authentication phase. The scheme proposed in [8] includes 4HP, 2BP, 2ADD, 11MUL operations. Therefore the computational cost of [8] is $4T_{HP} + 2T_{BP} + 2T_{ADD} + 11T_{MUL} \approx$ 352.66 ms. Secondly, the scheme proposed in [9] includes 4HP, 2BP, and 7MUL operations. Therefore the computational cost of [9] is $4T_{HP} + 2T_{BP} + 7T_{MUL} \approx 297.7$ ms. Next, the scheme proposed in [10] includes 2HP, 4BP, 41ADD, 4MUL operations. Therefore the computational cost of [10] is $2T_{HP} + 4T_{BP} + 41T_{ADD} + 4T_{MUL} \approx 227.48$ ms. Further, the scheme proposed in [7] includes 1HP, 2BP, and 8MUL operations. Therefore the computational cost of [7] is $T_{HP} + 2T_{BP} + 8T_{MUL} \approx 184.9$ ms. Lastly, the proposed protocol includes 2BP and 9MUL operations. Therefore the computational cost of our scheme is $2T_{BP} + 9T_{MUL} \approx 156.3$ ms. The total computational operations and computation costs of the various authentication methods compared are depicted in Tables 6 and 7. Clearly, from Fig. 6 it is evident that our protocol has the lowest computational overhead of all the alternatives. Thus the devised framework offers superior security and less computational overheads.

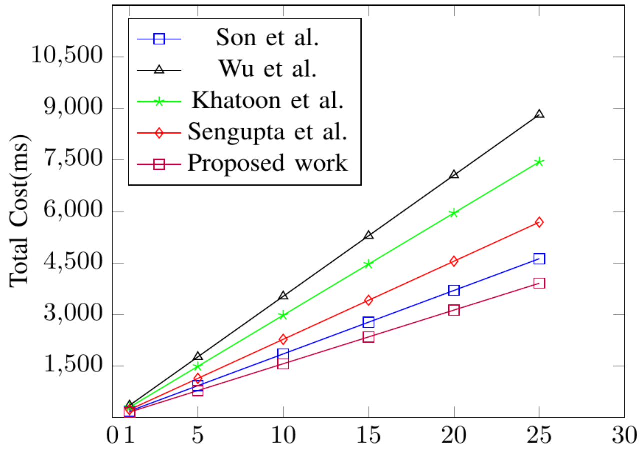

FIGURE 6. Computational costs comparison with respect to the number of authentications.

C. COMMUNICATION COST

This section evaluates the communication costs of the proposed framework and makes a comparison with [7], [8], [9], and [10]. The group elements, identity, timestamp, random number, and hash function output in the proposed scheme require 1024, 128, 32, 160, and 256 bits, respectively. The scheme proposed by Wu et al. [8] transmits two messages during the login and authentication phase. The first message in [8] is $(Id_i, R_{in}, R_{i1}, R_{si}, T_1)$, and the second message is $(Id_j, R_{jn}, R_{j1}, h_j)$. These messages contain two identities, a hash output, a timestamp, and five group elements of G_1. The total cost of communication is $2 \times 128 + 1 \times 256 + 1 \times 32 + 5 \times 1024 = 5664$ bits. Similarly, the scheme proposed by Khatoon et al. [9] also transmits two messages. The first message in [9] is $(R_i, T_i, Auth_i)$, and the second message is $(R_i, T_i, Auth_i)$. These messages contain two timestamps, two hash outputs, and two group elements of G_1. The total cost of communication is $2 \times 32 + 2 \times 256 + 2 \times 1024 = 2624$ bits. Further, the scheme proposed by Sengupta et al. [10] also transmits two messages $(CID_i, N_i, C_i, F_i, T_i)$ and $a, T_{ss})$. The messages contain two timestamps and five group elements of

TABLE 4. Security features.

Security features	Son et al. [7]	Wu et al. [8]	Khatoon et al. [9]	Sengupta et al. [10]	Proposed
Replay attack	✓	✓	✓	✓	✓
Offline password guessing attack	×	✓	✓	✓	✓
Privileged insider attack	×	✓	✓	✓	✓
Impersonation attack	×	✓	✓	✓	✓
Known session specific temporary information attack (KSSTIA)	×	×	-	×	✓
Perfect forward secrecy attack	✓	×	✓	✓	✓
Stolen smart device attack	×	-	-	-	✓
Session key computation attack	✓	✓	✓	✓	✓
Anonymity and untraceability attack	×	✓	×	×	✓
Mutual authentication	×	✓	✓	✓	✓
Data verification	✓	×	×	×	✓
Session key agreement	×	✓	✓	✓	✓
Formal security using BAN logic	✓	×	✓	✓	✓
Formal security using ROR random oracle model	×	✓	✓	×	✓

TABLE 5. Execution time of different cryptographic operations.

Operations	Symbols	Time (ms)
The map-to-point hash (MTP) operation	T_{HP}	42.1 ms.
The bilinear pairing (BP) operation	T_{BP}	17.4 ms.
The point addition (PA)	T_{ADD}	0.48 ms
The point-scalar multiplication (PM)	T_{MUL}	13.5 ms

TABLE 6. Computational operations.

Protocol	Computational operations
Son et al. [7]	$T_{HP} + 2T_{BP} + 8T_{MUL}$
Wu et al. [8]	$4T_{HP} + 2T_{BP} + 2T_{ADD} + 11T_{MUL}$
Khatoon et al. [9]	$4T_{HP} + 2T_{BP} + 7T_{MUL}$
Sengupta et al. [10]	$2T_{HP} + 4T_{BP} + 41T_{ADD} + 4T_{MUL}$
Proposed	$2T_{BP} + 9T_{MUL}$

TABLE 7. Computational costs comparison.

Protocol	Execution time (in milliseconds)
Son et al. [7]	184.9
Wu et al. [8]	352.66
Khatoon et al. [9]	297.7
Sengupta et al. [10]	227.48
Proposed	156.3

G_1. The total cost of communication is $2 \times 32 + 5 \times 1024 = 5184$ bits. Next, the scheme proposed by Son et al. [7] transmits messages $\{Q_r, PID_r, M_r, X_r, T_3\}$ and $\{U_s, L_s, T_4\}$. The messages include two timestamps, two hash outputs, and four group elements. The total cost of communication is $2 \times 256 + 2 \times 32 + 4 \times 1024 = 4672$ bits. In the authentication phase of our scheme, O_s and U_r have exchanged two messages. Both the messages $\{Q_r, PID_r, M_r, X_r, T_3\}$ and $\{U_s, L_s, T_4\}$ has a computational cost of 4672 bits which is equivalent to that of [7]. Table 8 demonstrates that, despite having a slightly higher communication cost than [9], and comparably lesser from [8] and [10] our scheme offers better security and functionality features and is more efficient.

TABLE 8. Communication costs comparison.

Protocol	Communication cost (bits)	No. of messages
Son et al. [7]	4672	2
Wu et al. [8]	5664	2
Khatoon et al. [9]	2624	2
Sengupta et al. [10]	5184	2
Proposed	4672	2

IX. CONCLUSION

In this article, we examined various design flaws and vulnerabilities of the scheme suggested in [7] in opposition to numerous cryptographic attacks, like user impersonation, KSSTIA, and offline password guessing attacks. By utilizing blockchain technology, we proposed an enhanced three-factor-based privacy-preserving authentication framework for the DT environment. The informal security analysis of the proposed scheme shows the efficiency and enhanced security against various wicked attacks. The mutual authentication and session key security is also ensured by performing the formal analysis of the proposed work using both the ROR Model and BAN logic. Moreover, compared to the competing existing works, the proposed method offers reduced computation costs, comparable communication costs, and superior security. Therefore, the proposed work is suitable for the DT environment.

In future, we would like enhance the proposed scheme with more efficiency in terms of communication, computational and storage costs while keeping the same security level. In addition, we would also like to develop a complete testbed experiment for practical aspects of the proposed scheme.

ACKNOWLEDGMENT

The authors would like to thank the associate editor and the seven anonymous reviewers of this manuscript in order to enhance its technical quality and presentation.

REFERENCES

[1] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*. Cham, Switzerland: Springer, 2017, pp. 85–113.

[2] B. Piascik, J. Vickers, D. Lowry, S. Scotti, J. Stewart, and A. Calomino, "Materials, structures, mechanical systems, and manufacturing roadmap," NASA, Washington, DC, USA, Tech. Rep. TA 12, 2012.

[3] H. Laaki, Y. Miche, and K. Tammi, "Prototyping a digital twin for real time remote control over mobile networks: Application of remote surgery," *IEEE Access*, vol. 7, pp. 20325–20336, 2019.

[4] H. Wang and J. Zhang, "Blockchain based data integrity verification for large-scale IoT data," *IEEE Access*, vol. 7, pp. 164996–165006, 2019.

[5] P. Wei, D. Wang, Y. Zhao, S. K. S. Tyagi, and N. Kumar, "Blockchain data-based cloud data integrity protection mechanism," *Future Gener. Comput. Syst.*, vol. 102, pp. 902–911, Jan. 2020.

[6] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, "Blockchain based data integrity service framework for IoT data," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2017, pp. 468–475.

[7] S. Son, D. Kwon, J. Lee, S. Yu, N.-S. Jho, and Y. Park, "On the design of a privacy-preserving communication scheme for cloud-based digital twin environments using blockchain," *IEEE Access*, vol. 10, pp. 75365–75375, 2022.

[8] T.-Y. Wu, Y.-Q. Lee, C.-M. Chen, Y. Tian, and N. A. Al-Nabhan, "An enhanced pairing-based authentication scheme for smart grid communications," *J. Ambient Intell. Humanized Comput.*, vol. 2021, pp. 1–13, Jan. 2021.

[9] S. Khatoon, S. M. M. Rahman, M. Alrubaian, and A. Alamri, "Privacy-preserved, provable secure, mutually authenticated key agreement protocol for healthcare in a smart city environment," *IEEE Access*, vol. 7, pp. 47962–47971, 2019.

[10] A. Sengupta, A. Singh, P. Kumar, and T. Dhar, "A secure and improved two factor authentication scheme using elliptic curve and bilinear pairing for cyber physical systems," *Multimedia Tools Appl.*, vol. 16, pp. 1–24, Jul. 2022.

[11] H. S. Grover and D. Kumar, "Cryptanalysis and improvement of a three-factor user authentication scheme for smart grid environment," *J. Reliable Intell. Environ.*, vol. 6, no. 4, pp. 249–260, Dec. 2020.

[12] M. Wazid, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Secure three-factor user authentication scheme for renewable-energy-based smart grid environment," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3144–3153, Dec. 2017.

[13] D. Kaur and D. Kumar, "Cryptanalysis and improvement of a two-factor user authentication scheme for smart home," *J. Inf. Secur. Appl.*, vol. 58, May 2021, Art. no. 102787.

[14] M. Shuai, N. Yu, H. Wang, and L. Xiong, "Anonymous authentication scheme for smart home environment with provable security," *Comput. Secur.*, vol. 86, pp. 132–146, Sep. 2019.

[15] Y. Chen, J. Martinez, P. Castillejo, and L. López, "A bilinear map pairing based authentication scheme for smart grid communications: PAuth," *IEEE Access*, vol. 7, pp. 22633–22643, 2019.

[16] M. Nikooghadam and H. Amintoosi, "Cryptanalysis of Khatoon et al.'s ECC-based authentication protocol for healthcare systems," 2019, *arXiv:1906.08424*.

[17] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. Int. Workshop Public Key Cryptogr.* Cham, Switzerland: Springer, 2005, pp. 65–84.

[18] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, Feb. 1990.

[19] A. K. Das and B. Bruhadeshwar, "An improved and effective secure password-based authentication and key agreement scheme using smart cards for the telecare medicine information system," *J. Med. Syst.* vol. 37, no. 5, p. 9969, 2013.

[20] S. Chatterjee and A. K. Das, "An effective ECC-based user access control scheme with attribute-based encryption for wireless sensor networks," *Secur. Commun. Netw.*, vol. 8, no. 9, pp. 1752–1771, 2015.

[21] D. Mishra, A. K. Das, and S. Mukhopadhyay, "A secure and efficient ECC-based user anonymity-preserving session initiation authentication protocol using smart card," *Peer-Peer Netw. Appl.*, vol. 9, no. 1, pp. 171–192, Jan. 2016.

[22] S. Mandal, B. Bera, A. K. Sutrala, A. K. Das, K. R. Choo, and Y. Park, "Certificateless-signcryption-based three-factor user access control scheme for IoT environment," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3184–3197, Apr. 2020.

[23] M. Wazid, S. Thapliyal, D. P. Singh, A. K. Das, and S. Shetty, "Design and testbed experiments of user authentication and key establishment mechanism for smart healthcare cyber physical systems," *IEEE Trans. Netw. Sci. Eng.*, early access, Mar. 29, 2022, doi: 10.1109/TNSE.2022.3163201.

[24] B. Bera, A. K. Das, W. Balzano, and C. M. Medaglia, "On the design of biometric-based user authentication protocol in smart city environment," *Pattern Recognit. Lett.*, vol. 138, pp. 439–446, Oct. 2020.

[25] A. K. Das, N. R. Paul, and L. Tripathy, "Cryptanalysis and improvement of an access control in user hierarchy based on elliptic curve cryptosystem," *Inf. Sci.*, vol. 209, pp. 80–92, Nov. 2012.

[26] S. Zeadally, A. K. Das, and N. Sklavos, "Cryptographic technologies and protocol standards for Internet of Things," *Internet Things*, vol. 14, Jun. 2021, Art. no. 100075.

[27] A. K. Das, S. Zeadally, and D. He, "Taxonomy and analysis of security protocols for Internet of Things," *Future Gener. Comput. Syst.*, vol. 89, pp. 110–125, Dec. 2018.

[28] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services," *IEEE Access*, vol. 5, pp. 25808–25825, 2017.

[29] S. Banerjee, V. Odelu, A. K. Das, J. Srinivas, N. Kumar, S. Chattopadhyay, and K.-K.-R. Choo, "A provably secure and lightweight anonymous user authenticated session key exchange scheme for Internet of Things deployment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8739–8752, Oct. 2019.

[30] B. Bera, A. K. Das, and A. K. Sutrala, "Private blockchain-based access control mechanism for unauthorized UAV detection and mitigation in Internet of Drones environment," *Comput. Commun.*, vol. 166, pp. 91–109, Jan. 2021.

[31] S. Aheleroff, X. Xu, R. Y. Zhong, and Y. Lu, "Digital twin as a service (DTaaS) in industry 4.0: An architecture reference model," *Adv. Eng. Informat.*, vol. 47, Jan. 2021, Art. no. 101225.

[32] C. Lai, M. Wang, and D. Zheng, "SPDT: Secure and privacy-preserving scheme for digital twin-based traffic control," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2022, pp. 144–149.

[33] Y. Liu, L. Zhang, Y. Yang, L. Zhou, L. Ren, F. Wang, R. Liu, Z. Pang, and M. J. Deen, "A novel cloud-based framework for the elderly healthcare services using digital twin," *IEEE Access*, vol. 7, pp. 49088–49101, 2019.

[34] J. Liu, L. Zhang, C. Li, J. Bai, H. Lv, and Z. Lv, "Blockchain-based secure communication of intelligent transportation digital twins system," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22630–22640, Nov. 2022.

[35] S. Huang, G. Wang, Y. Yan, and X. Fang, "Blockchain-based data management for digital twin of product," *J. Manuf. Syst.*, vol. 54, pp. 361–371, Jan. 2020.

[36] S. Vairavasundaram, K. Kotecha, L. Ravi, G. Selvachandran, and A. Abraham, "Blockchain-based trust mechanism for digital twin empowered industrial Internet of Things," *Future Gener. Comput. Syst.*, vol. 141, pp. 16–27, Apr. 2023.

[37] C. Wang, Z. Cai, and Y. Li, "Sustainable blockchain-based digital twin management architecture for IoT devices," *IEEE Internet Things J.*, early access, Feb. 23, 2022, doi: 10.1109/JIOT.2022.3153653.

[38] J. Lee, S. Yu, M. Kim, Y. Park, and A. K. Das, "On the design of secure and efficient three-factor authentication protocol using honey list for wireless sensor networks," *IEEE Access*, vol. 8, pp. 107046–107062, 2020.

[39] P. Bagga, A. K. Das, M. Wazid, J. J. Rodrigues, K.-K. R. Choo, and Y. Park, "On the design of mutual authentication and key agreement protocol in Internet of Vehicles-enabled intelligent transportation system," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1736–1751, Jan. 2021.

[40] A. T. B. Jin, D. N. C. Ling, and A. Goh, "BioHashing: Two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognit.*, vol. 37, no. 11, pp. 2245–2255, Apr. 2004.

[41] A. K. Das, "A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor," *Int. J. Commun. Syst.*, vol. 30, no. 1, p. e2933, Jan. 2017.

[42] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo, "Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things," *IEEE Internet Things*, vol. 5, no. 4, pp. 2884–2895, Aug. 2017.

[43] M. Castro and B. Liskov, "Practical Byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, 2002.

[44] J. Ryu, S. Son, J. Lee, Y. Park, and Y. Park, "Design of secure mutual authentication scheme for metaverse environments using blockchain," *IEEE Access*, vol. 10, pp. 98944–98958, 2022.

[45] Y. Cho, J. Oh, D. Kwon, S. Son, S. Yu, Y. Park, and Y. Park, "A secure three-factor authentication protocol for E-governance system based on multiserver environments," *IEEE Access*, vol. 10, pp. 74351–74365, 2022.

[46] J. Lee, G. Kim, A. K. Das, and Y. Park, "Secure and efficient honey list-based authentication protocol for vehicular ad hoc networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2412–2425, Jul. 2021.

GARIMA THAKUR received the B.Sc. degree from Himachal Pradesh University, Shimla, and the M.Sc. degree from the Central University of Himachal Pradesh, Dharamshala, where she is currently pursuing the Ph.D. degree. Her research interests include authentication, post-quantum cryptography, the IoT, and blockchain technology.

PANKAJ KUMAR received the M.Sc. degree from CCS University Meerut, India, and the Ph.D. degree from Galgotias University, in 2005 and 2020, respectively. He has been an Assistant Professor with the Srinivasa Ramanujan Department of Mathematics, Central University of Himachal Pradesh, Dharamshala, Himachal Pradesh. He has published over 40 academic research articles on information security and privacy preservation. His current research interests include cryptography, blockchain, wireless network security, information theory, and network coding.

DEEPIKA received the B.Sc. degree from Himachal Pradesh University, Shimla, India, and the M.Sc. degree from the Central University of Himachal Pradesh, Dharamshala, India, where she is currently pursuing the Ph.D. degree. Her research interests include digital signature, authentication, and blockchain technology.

SRINIVAS JANGIRALA (Member, IEEE) received the B.Sc. and M.Sc. degrees from Kakatiya University, India, in 2003 and 2008, respectively, the M.Tech. degree from IIT Kharagpur, in 2011, and the Ph.D. degree from the Department of Mathematics, IIT Kharagpur, in 2017. He is currently an Associate Professor with the Jindal Global Business School, O. P. Jindal Global University, Haryana, India. Prior to this, he was a Research Assistant with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology (IIIT), Hyderabad, India. His research interests include blockchain technology and applications, information security, cryptocurrency, and supply chain. He has authored 30 papers in international journals and conferences in his research areas.

ASHOK KUMAR DAS (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering, the M.Tech. degree in computer science and data processing, and the M.Sc. degree in mathematics from IIT Kharagpur, India. He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, IIIT Hyderabad, India, and also a Visiting Faculty with the Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA, USA. His current research interests include cryptography, system and network security including security in smart grids, the Internet of Things (IoT), the Internet of Drones (IoD), the Internet of Vehicles (IoV), cyber-physical systems (CPS) and cloud computing, intrusion detection, blockchain, and AI/ML security. He has authored over 335 papers in international journals and conferences in the above areas, including over 290 reputed journal articles. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He has been listed in the Web of Science (Clarivate) Highly Cited Researcher 2022 in recognition of his exceptional research performance. He was/is on the editorial boards of IEEE Systems Journal, *Journal of Network and Computer Applications* (Elsevier), *Computer Communications* (Elsevier), *Journal of Cloud Computing* (Springer), *Cyber Security and Applications* (Elsevier), *IET Communications*, *KSII Transactions on Internet and Information Systems*, and *International Journal of Internet Technology and Secured Transactions* (Inderscience). He severed as one of the Technical Program Committee Chairs for the first International Congress on Blockchain and Applications (BLOCKCHAIN'19), Ávila, Spain, in June 2019; the International Conference on Applied Soft Computing and Communication Networks (ACN'20), Chennai, India, in October 2020; and the Second International Congress on Blockchain and Applications (BLOCKCHAIN'20), L'Aquila, Italy, in October 2020. His Google Scholar H-index is 75 and i10-index is 213 with over 15,500 citations.

YOUNGHO PARK (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering, Kyungpook National University, Daegu, South Korea, in 1989, 1991, and 1995, respectively. From 1996 to 2008, he was a Professor with the School of Electronics and Electrical Engineering, Sangju National University, South Korea. From 2003 to 2004, he was a Visiting Scholar with the School of Electrical Engineering and Computer Science, Oregon State University, USA. He is currently a Professor with the School of Electronics Engineering, Kyungpook National University. His research interests include information security, computer networks, and multimedia.