Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

An ECC-based mutual data access control protocol for next-generation public cloud

Naveed Khan[1], Zhang Jianbiao[1,2†], Huhnkuk Lim[3*†], Jehad Ali[4†], Intikhab Ullah[5†], Muhammad Salman Pathan[6†] and Shehzad Ashraf Chaudhry[7,8†]

Abstract

Through the broad usage of cloud computing and the extensive utilization of next-generation public clouds, people can share valuable information worldwide via a wireless medium. Public cloud computing is used in various domains where thousands of applications are connected and generate numerous amounts of data stored on the cloud servers via an open network channel. However, open transmission is vulnerable to several threats, and its security and privacy are still a big challenge. Some proposed security solutions for protecting next-generation public cloud environments are in the literature. However, these methods may not be suitable for a wide range of applications in a next-generation public cloud environment due to their high computing and communication overheads because if security protocol is strengthened, it inversely impacts performance and vice versa. Furthermore, these security frameworks are vulnerable to several attacks, such as replay, denial-of-service (DoS), insider, server spoofing, and masquerade, and also lack strong user anonymity and privacy protection for the end user. Therefore, this study aims to design an elliptic curve cryptographic (ECC) based data access control protocol for a public cloud environment. The security mechanism of the proposed protocol can be verified using BAN (Burrows-Abadi-Needham) logic and ProVerif 2.03, as well as informally using assumptions and pragmatic illustration. In contrast, in the performance analysis section, we have considered the parameters such as the complexity of storage overheads, communication, and computation time. As per the numerical results obtained in the performance analysis section, the proposed protocol is lightweight, robust, and easily implemented in a practical next-generation cloud computing environment.

Keywords Next-Generation Public Cloud, Pi-Calculus, ECC, Authentication, ProVerif

†Zhang Jianbiao, Huhnkuk Lim, Jehad Ali, Intikhab Ullah, Muhammad Salman Pathan and Shehzad Ashraf Chaudhry contributed equally to this work.

*Correspondence:
Huhnkuk Lim
slow63347@gmail.com
[1] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
[2] Beijing Key Laboratory of Trusted Computing, Beijing University of Technology, Beijing 100124, China
[3] Department of Computer Engineering, Hoseo University, Asan-si 31499, South Korea
[4] Department of AI Convergence Network, Ajou University, Suwon 16499, South Korea
[5] Department of Computer Science, Shaheed Benazir Bhutto University, Sheringal, Dir Upperl 18050, Pakistan
[6] Department of Computer Science, Maynooth University, Maynooth W23 A3HY, Ireland
[7] Department of Software Engineering, Faculty of Engineering and Architecture, Nisantasi University, Istanbul 34398, Turkey
[8] Department of Computer Science and Information Technology, College of Engineering, Abu Dhabi University, Abu Dhabi, UAE

Introduction

With the rapid development of high-speed internet, cloud computing has become an essential factor in the information and communication technology (ICT) industry. Next-generation cloud computing provides internet services without exposing the end user's physical location and system configuration. Cloud computing moves data and computing away from laptops, portable PCs, and desktops into large data centres. It enables better use of distributed resources, combining them to achieve higher throughput and solve huge-scale computation problems [1, 2]. The word "cloud" came from telecommunications when telecom companies started offering Virtual Private Network (VPN) services. Before VPN, these companies provided point-to-point data over circuits that squandered bandwidth. However, using VPN services, they managed to balance the overall network utilization [2]. According to John McCarthy, way back in the 1960s, "computation may someday be organized as a public utility" [3]. In his book, "The Challenge of the Computer Utility" [4], Douglas Parkhill describes how the characteristics of cloud computing were first investigated in 1966. Using next-generation cloud computing, we can access a shared pool of configurable computing resources [2]. These resources can be storage, applications, networks, servers, and services that can be managed with little help from the cloud service provider (CPS) [5]. Cloud computing has many benefits, but security and privacy-related problems must be addressed before migrating to the cloud. Security issues exist at both ends of next-generation public cloud computing, such as users and cloud service providers [6]. Many researchers have proposed different authentication schemes, but these schemes have high communication and computation cost. These methods, however, may not be suitable for a wide range of applications in a next-generation public cloud environment due to their high computing and communication overheads. Furthermore, these security frameworks are susceptible to a wide range of attacks, including, but not limited to, man-in-the-middle (MITM), replay, denial-of-service (DoS), insider, server spoofing, masquerade threats, and many others. Also, it does not provide adequate safeguards to ensure the privacy and anonymity of its users. Consequently, this work aims to develop an ECC-based authentication system for the privacy of a next-generation public cloud environment. ECC is a public-key cryptography scheme based on the algebraic structure of the elliptic curve over finite fields. The key size of ECC-based protocols is smaller than other non-ECC cryptographic techniques. Thus, it reduces the storage and transmission requirements. For example, if the RSA public key uses 3072 bits to secure data, the ECC provides the same security in just 256 bits. Therefore, the ECC reduces packet overheads, power consumption and removes needless computation power [7]. Finally, the performance analysis section considered the complexity of storage overheads, communication, and computation costs. In the end, the analysis section shows that the proposed scheme is lightweight, robust, and easily implemented in a practical next-generation public cloud computing environment.

Motivation and contributions

With the proliferation of next-generation public cloud computing and the integration of the Internet of Things (IoT), the most difficult challenge is the authentication of a remote user over an insecure channel. Few security applications provide anonymity from adversaries but cannot be offered in the case of a next-generation public cloud. The author [8] utilized a discrete logarithm problem to authenticate three-party authentication. According to our analysis, the protocol [8] has high computation and communication overheads that are unsuitable for energy and resource constraint devices such as mobile and IoT and also insecure against potential threats, which was a prime motivation for us to design a new security scheme. Other main contributions of this research paper are as under:

1. The proposed protocol uses lightweight elliptical curve cryptography (ECC) public key technique to resist potential attacks.
2. The proposed protocol is verifiably protected in the BAN logic against the hardness assumptions of the elliptic curve discrete logarithm problem and the elliptic curve computational Diffie-Hellman problem.
3. The proposed protocol is secure, based on analyzing the automated verification software toolkit ProVerif 2.03.
4. The proposed protocol will have lower communication costs, computation complexity, and less storage overhead than the existing protocols.
5. The proposed protocol significantly balances the performance and security measures often lacking in existing schemes.

Paper organization

This article is organized as follows: Section "Preliminaries" provides a detailed examination of the preliminary topics. The literature review is presented in Section "Literature review". Section "Proposed scheme" focuses on the proposed scenario, while Section "Security analysis" examines the security analysis. The performance of the framework is explored in Section "Performance analysis". Finally, Section "Conclusion" concludes the paper.

Preliminaries

This section discusses, defines, and illustrates the key terms used in this research study. The basic concepts and ideas are necessary to improve cryptography and security, which are not exhaustively considered.

Hash function

A hash function is a mathematical function used to check the message's integrity [5]. Moreover, the hash function converts a numerical input value into a compressed numerical value. A hash function's output value, on the other hand, is always of a fixed length, such as $h(0,1)^* \epsilon Zq^*$ produces fixed size out $M = h(Str)$, where Str is a random size input string [9]. It is simple and easy to compute $M = h(Str)$ if Str is given, although it is impractical to figure out Str if $M = h(Str)$ is specified. Furthermore, the size of a hash value is significantly less than the original message because the hash function sometimes refers to the compression function. Therefore, converting messages to hash values is easy. However, the actual message is very hard to calculate from the hash value. Moreover, when an exact message is converted η times, the hash function values remain the same. Therefore, the hash function could not be successfully performed when the two messages have the same hash value. However, it is not the case in general.

Elliptic curve cryptography (ECC)

ECC is more efficient than Diffie Hellman, DSA, and RSA cryptographic protocols [7, 10]. The ECC is defined by an elliptic curve $Eq(a,b) : y^2 = x^3 + ax + b \ mod \ q$, where a and b belong to $Z_q\{a, b \epsilon Z_q^*\}$. Furthermore, a and b selection must be carefully chosen to satisfy $4a^3 + 27b \ mod \ q \neq 0$, where q represents a prime number, and the length of q is equal and greater than 160 bits $\{|q| \geq 160 \ bits\}$. The ECC has two operations: the ECC point of multiplication and the ECC point of addition. ECC is symmetric about the $x - axis$. Therefore, drawing a line on the graph takes a maximum cut of 3 points. Let $E_q(a,b)$ be the point on the elliptic curve, consider the equation $Q = KP$ where Q, P point on the curve and $K \leqslant n$. if K and P are given, it should be easy to find Q, but it is complicated to find K if we know Q and P.

System model

The system model presented in this subsection consisted of the end-user and next-generation public cloud servers (NG-PCS). The next-generation public cloud server plays a crucial role in our system model. It provides connectivity, data access, storage facilities, data sharing, cookies, and other security services for all its users. It is worth mentioning here that the next-generation public cloud server is considered a trusted entity because, without declaring it trusted, it will degrade the overall system. The end user alone cannot be trusted because it can degrade system credibility. In order to access the next-generation public cloud server, the user sends a request message. The next-generation public cloud server sends a challenge message back to the user to check the authenticity of a user. After receiving the response message from the user, the next-generation public server starts the session with the user if the user is legit. The detailed procedure is explained in the proposed protocol. Our system model is shown in Fig. 1.

Threat model

Dolev and Yao first demonstrated the DY threat model [11]. We have extended the threat model used in [1, 12] by adopting a solid adversary (∀). According to this model, any threat to the system can be analyzed and examined before operationalizing it for the real-world environment. So, keeping this view, all possible threats to our system include:

1. An ∀ intercepts the communication between a user and the next-generation public cloud server.
2. An ∀ captures the open network channel data for possible insertion, deletion, and modification.
3. The ∀ can dynamically monitor the broadcast message among legal peers.
4. An ∀ can act as a legal participant and launch masquerading, impersonation, and man-in-the-middle attacks.
5. The communication medium between the user and the next-generation public cloud server is secure. However, the ∀ can intrude into the insecure communication medium between the user and the next-generation public cloud server.
6. The identity of the next-generation public cloud server is publicly announced.
7. The ∀ cannot extract the next-generation public cloud server's secret key SK_{pcs}.
8. Other system threats are privacy, server spoofing, insider, and session key threats.

Security goals

The security requirement goals were introduced by Canetti [13]. The increasing usage of next-generation cloud computing increases the probability of being attacked remotely. The cloud service provider or the

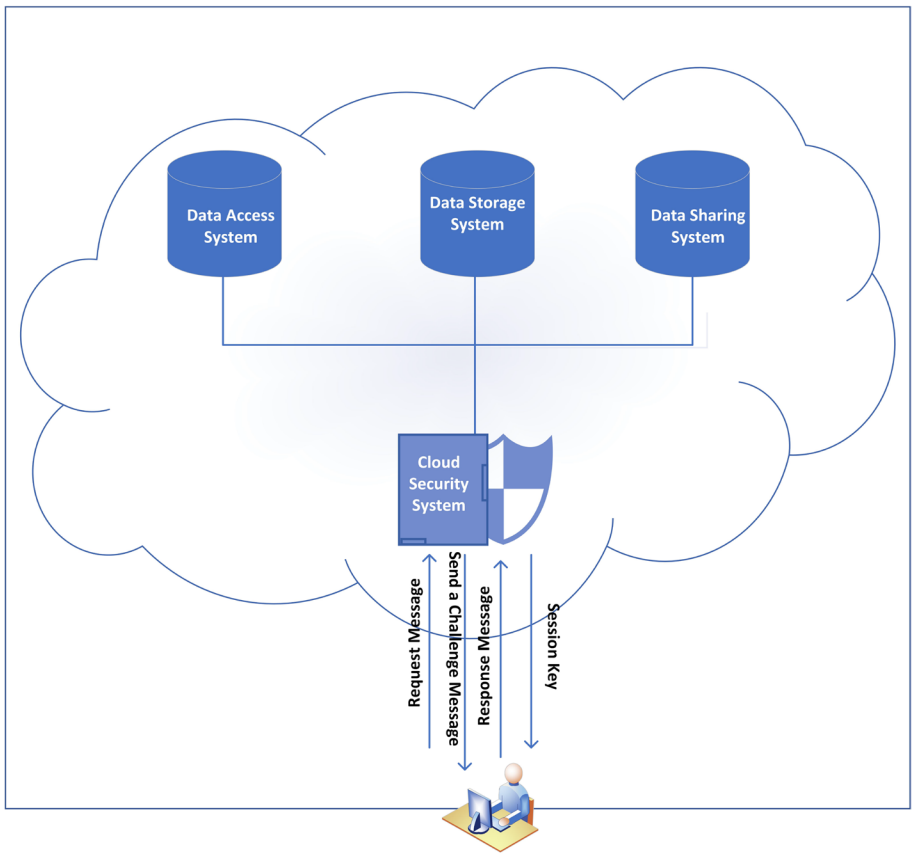

Fig. 1 System model

user can be compromised in next-generation cloud computing. Therefore, the attacker can take control of user data without having physical access to the server. Moreover, remote access to the cloud can compromise sensitive information, and the primary cause of unauthorized access to the communication may be replayed, modified, spoofed, or eavesdropped. Thus, all communication peers must be authenticated to prevent unauthorized access to secure transmitted information. Therefore, our proposed scheme aims to achieve the following security goals (SGs) and compares SG's existing schemes in the performance section.

SG 1. Resists Offline Password Guessing Attacks: The ∀ can guess the user's password and gets access to sensitive data of the next-generation public cloud server. However, the protocol should be strong enough to prevent this attack.

SG 2. Free from De-synchronization Attacks: The ∀ tries to update or synchronize something on both peers, such as the ∀ trying to update a user's password or block communication between the next-generation public cloud server and the user. The protocol should be capable of resisting de-synchronization attacks.

SG 3. Provision of Key Agreement: The user and next-generation public cloud server authenticate using secure values and integers to communicate securely.

SG 4. Spoofing Attacks: The ∀ falsifying data and trying to impersonate a legit user to access the next-generation public cloud server. Therefore, the proposed scheme should be strong enough to prevent spoofing attacks.

SG 5. Resists Insider Attack:∀ tries to pretend to be a legitimate user or attempts to access the user-sensitive data. Therefore, the protocol must prevent insider attacks.

SG 6. Perfect Message Authentication: The ∀ trying to send unverified messages to the next-generation public cloud server or user to gain access to the communication. Therefore, the scheme should prevent the next-generation public cloud server and the user from falsifying messages.

SG 7. User Anonymity: The scheme should be capable of maintaining user anonymity if the ∀ is trying to capture an authentication message.

SG 8. Mutual Authentication: The scheme should mutually authenticate the next-generation public cloud

server and user to prevent the ∀ from pretending to be one of the peers.

SG 9. Replay attacks: An ∀ captures the previous session and tries to repeat or delay the communication to confuse peers. The proposed scheme should be capable of preventing replay attacks.

SG 10. Impersonate attack: The ∀ tries to use the user identity or password to impersonate the user to access the next-generation public cloud server. However, the protocol should identify the legit user to prevent impersonation attacks.

SG 11. Provide Password Revocation/Changing: If the user forgets his/her password, the proposed scheme should provide a password revocation/changing option to access the next-generation public cloud server.

Literature review

Next-generation public cloud computing offers unlimited access to resources over the internet [14]. It is because of the availability of high-speed internet, more individuals and organizations are outsourcing their data to next-generation cloud servers for later access via the internet, and it reduces the burden on local storage. Although, different sources continuously generate a large amount of data that are outsourced to next-generation public cloud servers [15, 16]. However, securing outsourced data in the cloud is imperative for data owners [17]. In addition, the authentication protocols enable users to access these services through the remote servers over an insecure network. Furthermore, in this part of the literature, we discuss various schemes that have security vulnerabilities and high computation and communication overheads. In 1981, the author [18] proposed the first authentication protocol, which uses a username and password to secure a user's access to a server. Nevertheless, there was a drawback in the protocol because it maintained a password table. As a result, the adversary can intercept the password, perform a replay attack, and successfully log into the server. In 1990, a more secure two-factor authentication protocol was proposed by [19]. The two-factor authentication protocols use a username, password, and a smartcard. The topic of smartcard loss attacks has recently been brought up in the authentication schemes. Three-factor authentication combines a username, password, and a smartcard, which is a more secure method to access remote servers. Traditional two-factor authentication methods are only used for a single server environment. On the other hand, commercial services are based on a multi-server environment. Thus, the traditional authentication protocols do not provide untraceability and anonymity. Therefore, the author [20] proposed a three-factor authentication protocol; however, according to [21], the protocol proposed in [20] cannot provide user anonymity and is vulnerable to impersonation attacks. Furthermore, a multi-server environment scheme based on three-factor authentication was proposed by [20, 22]. However, according to [23], the protocol [22] is vulnerable to user impersonation attacks. The author [24] improves the security drawbacks of the scheme [25]. Moreover, the protocol [24] is vulnerable to insider and smartcard loss attacks. Therefore, the author [26] improves and solves the security vulnerabilities of the protocol [24]. The author [27] cryptanalysis the scheme [28] and find out security vulnerabilities such as the scheme cannot provide user anonymity. These security vulnerabilities were solved by [29]. The protocol designed by [30] stores the user's public keys on the server side, and this practice leads to man-in-the-middle attacks. Additionally, the author's [31] proposed scheme that cannot provide user untraceability and suffers from insider, server impersonation, and man-in-the-middle attacks. Moreover, the scheme [32] exposed the security vulnerabilities in protocols [26, 33]. Therefore, some researchers pay attention to network security, such that [34] proposed a scheme that places a network inspection detection system to verify packets received by the cloud. However, their approach has some drawbacks in its performance. Furthermore, [35] focuses on virtual network security to solve security issues between firewalls and virtual machines. However, against malicious external traffic, the scheme is powerless. On the other hand, the DDOS protection service [36, 37] and the intrusion detection system's importance were presented by [38]. Moreover, cloud computing offers multi-type network-based services. Thus, a single network security service will not fulfill the network security requirement. Because the cloud is a multi-tenant environment, the security and privacy challenges differ from those encountered in traditional computing environments [39]. According to [40], the existing three-factor authentication schemes have too much communication and computation cost. In addition, they do not have a dynamic revocation mechanism. The scheme proposed by [41] does not establish a session key, and the communication cost is also very high. According to [23], the protocol used by [42] suffers from temporary session information attacks. So far, different approaches have been used to authenticate remote users and eliminate the risk of cyberattacks. The authors [43] proposed an authentication scheme to provide secure authentication for the telecare medicine information system (TMIS). The protocol [43] used XOR, a one-way hash function, and a one-time password. The authors [44, 45] proposed an authentication scheme based on ECC in smart grid

environments. The author [46] proposed an ECC-based authentication scheme for telecare medical information systems. Although, the scheme is robust, but vulnerable to offline password guessing attacks, DoS attacks, and user impersonation attacks and cannot provide perfect forward secrecy and anonymity. Another three-factor authentication protocol is proposed by [47]. However, according to [48], the scheme [47] is vulnerable to insider attacks and cannot provide anonymity. Furthermore, the author [48] also claimed that there is a flaw in the scheme [47] password update phase as well. Moreover, accurate authentication of users can prevent forgery attacks. However, to avoid forgery attacks in next-generation public cloud computing, the client and the cloud server must authenticate using mutual or one-way authentication. Although many authentication protocols are proposed in [33, 49–51], but most of these schemes lack mutual authentication. These existing protocols are also vulnerable to MITM, impersonation, synchronization, and playback attacks [52]. According to [53], mutual authentication is essential to determine if the communication between two parties is genuine. Recently, researchers have developed authentication schemes based on lattice-based and Identity-Based Encryption (IBE) [54]. IBE is a two-factor security protection mechanism proposed by the authors [55]. In the IBE protocol, the sender only needs to know the receiver's identity to which it wants to send data, and no other information is required. The sender transmits data to the cloud, where the receiver can download the necessary data when needed. To decrypt the data, the receiver needs two things. The first one is the secret key, and the second is a unique personal security device. Hence, it is impossible to decrypt the ciphertext without these two devices. Hereafter, the unique personal device can be revoked if it gets lost. In both cases, the user data on the next-generation public cloud server is highly vulnerable to access by an adversary. Furthermore, in the authentication process, deep learning and neural network are also used [56]. Deep learning is a type of machine learning in which algorithms are used to learn from large datasets [57]. Neural networks are a type of artificial neural network (ANN) made up of interconnected layers of neurons that use inputs to generate predictions or decisions [58, 59]. Both of these technologies are used in authentication systems to assist in identifying and authenticating users, detecting anomalies and fraud, and improving system security. Deep learning can recognize patterns and classify data, whereas neural networks can detect patterns and anomalies, as well as recognize faces and fingerprints. Although, elliptic curve cryptography (ECC) is a prominent asymmetric cryptographic scheme that can provide security like the RSA technique, with a smaller key size and lightweight nature. However, according to [60] that the scheme used in [61] and [62] suffers from no anonymity, DoS, reply, masquerade and impersonation attacks. Meanwhile, in the scheme [62], information can be easily intercepted and injected by an adversary over a public channel. However, according to [63, 64], it is impossible for anyone to inject false information and break the credentials of the session shared key by using the Elliptic Curve computation Diffie-Hellman technique [63] and Elliptic Curve Discrete Logarithmic problem [64]. As a result, according to the author [65], the session key is not secure between different peers in the [66], and the scheme is vulnerable to masquerade attacks. Hence, a protocol has been proposed in [66] to overcome [65] scheme issues by using a robust protocol based on ECC by keeping a point at infinity on the curve. Through this technique, the attacker cannot challenge the legitimacy of the peers. Meanwhile, the author [67] is trying to solve the DoS attack in the scheme [65]. Finally, the abovementioned schemes have security vulnerabilities, high communication, and computation overheads. Therefore, it is not suitable for resources and energy constraints devices. Thus, we are designing a lightweight authentication protocol for next-generation public cloud computing to achieve lower computation and communication overhead while not compromising on security. We also illustrate a comprehensive literature review in Table 1.

Proposed scheme

This section presents the proposed ECC-based mutual data access control protocol for the next-generation public cloud server. There are 4 phases of our proposed protocol, i.e., setup, registration, key agreement, and password change. Each of these phases is described below, and Table 2 shows the symbols used to describe them.

Setup procedure

This phase of our proposed protocol is accomplished in the following ways.

1. At the start, the Public Cloud Server (PCS) picks a non-singular elliptic curve $E(F_q)$.
2. Select Point P from the curve of ECC $(E/F_q) \rightarrow P$
3. Select collision-free one-way hash functions $h_1(.) \rightarrow \{0,1\}^*, h_2(.) \rightarrow \{1,0\}^*$
4. Select a secret value of *PCS*, namely s.
5. Compute a master secret key $MSK_{PCS} = P.s$

Table 1 Comprehensive literature review

Schemes	Limitations
[61]	The scheme is vulnerable to reply, masquerade, DoS, impersonation attacks and unable to provide anonymity.
[47]	The protocol cannot provide anonymity and is vulnerable to insider attacks.
[46]	The scheme is vulnerable to offline password guessing, DoS, user impersonation attacks and cannot provide perfect forward secrecy and anonymity.
[62]	The scheme unable to provide anonymity and vulnerable to masquerade, DOS, reply, impersonation attacks.
[31]	The protocol cannot provide user untraceability and suffer from insider attacks.
[24]	The protocol is vulnerable to insider attacks, and smartcard lost attacks.
[22]	The scheme suffers from user impersonation attacks.
[20]	The protocol cannot provide user anonymity and vulnerable to impersonation attacks.
[41]	The scheme is vulnerable to offline password guessing, spoofing, impersonation attacks and unable to provide mutual authentication.
[42]	The scheme is unable to provide mutual authentication, perfect message authentication and vulnerable to offline password guessing, and de-synchronization attacks.
[40]	The scheme is vulnerable to de-synchronization, spoofing attacks, and unable to provide perfect message authentication.
[68]	The protocol is vulnerable to spoofing and de-synchronization attacks.
[69]	The scheme is vulnerable to spoofing attacks and unable to provide perfect message authentication.
[70]	The scheme is unable to provide perfect message authentication and vulnerable to de-synchronization, spoofing and impersonation attacks.
[71]	The protocol is vulnerable to offline password guessing attack, de-synchronization attacks and unable to provide mutual authentication.
[7]	The protocol is vulnerable to offline password guessing, de-synchronization attacks and unable to provide perfect forward secrecy.

Table 2 Common symbols

Symbol	Description	Symbol	Description
u	A user of the system	PCS	Public Cloud Server
ID_u	Identity of user	F_q	The Prime finite fields
$E/(Fq)$	Elliptic curve over F_q	Z_q^*	Additive group of order q
ID_{PCS}	Identity of public cloud server	SK_u, SK_{PCS}	Session key
P	Elliptic Curve Point	G	Additive ECC group
g	The base point of order G	$h(.)$	One-way hash function
$\bigoplus$	XOR operator	$\|$	Concatenation Function
PW_u	User Password	T	Timestamp

6. The *PCS* selects g, the base point of an order G.
7. Finally, the public cloud server publishes $g, G, h_1(.), h_2(.)$.

Registration procedure

This phase of our proposed protocol is accomplished in the following ways.

1. User, u will register itself with the public cloud server in the registration phase. The u chooses identity ID_u password PW_u and a random number $r_u \epsilon Z_q^*$. Furthermore, the u computes $PID_u = h(ID_u \parallel r_u \parallel PW_u)$ and sends $< ID_u, PID_u >$ over a secure channel.
2. After receiving, the message from u, the public cloud server generates a random number $r_{PCS} \epsilon Z_q^*$ and computes $V_1 = ID_u \bigoplus r_{PCS}, V_2 = h(ID_u \parallel V_1 \parallel r_{PCS}) \bigoplus MSK_{PCS}$, and $V_3 = V_2 \bigoplus PID_u$. After computation, the public cloud server sends $< V_1, V_3, G, g, h(.) >$ message towards u over a secure channel.
3. On receiving $< V_1, V_3, G, g, h(.) >$ message, the user u calculates $S_1 = r_u \bigoplus h(ID_u \parallel PID_u)$, $S_2 = h(PID_u \parallel S_1), S_3 = r_u \bigoplus h(ID_u \parallel V_1)$ and saves $\{S_1, S_2, S_3, g, G, h(.)\}$ in its memory as shown in Fig. 2.

Key-agreement procedure

After successful registration, the user sends an entry request to the next-generation public cloud server for mutual authentication and cross-verification.

1. The user provides an identity ID_u password PW_u and compute $r_u^* = r_u^* = S_1 \bigoplus h(ID_u \| PID_u)$, $PID_u^{/} = h(ID_u \| r_u^* \| PW_u), S_2^{/} = h(PID_u^{/} \| S_1)$ and verify $S_2^{/} ? = S_2$. If it is validated, compute it further; otherwise, suspend it. Now, the user u generates another random number $r_{u1} \epsilon Z_q^*$ and computes $V_2^* = V_3 \bigoplus PID_u^{/}, V_1^* = h(ID_u \| V_1) \bigoplus r_{u1}, C_{2=h}(ID_u \| r_u \| V_1^*)$, $K_{u1} = h(ID_u \| r_u \| S_3)$ and selects timestamp T_{LA1}, compute $N = h(ID_u \| V_1^* \| C_2 \| V_2^* \| T_{LA1})$ and sends $< S_3, V_1^*, C_2, N, T_{LA1} >$ message toward cloud server over a public channel.

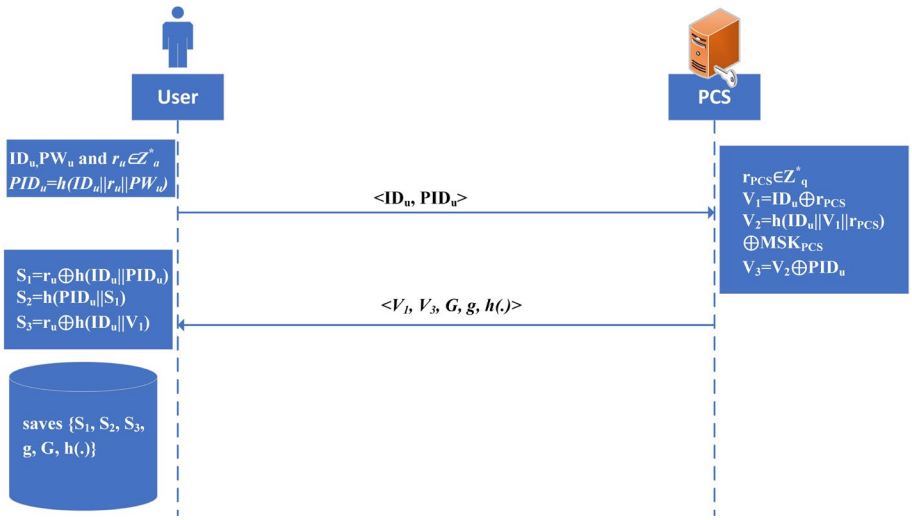

Fig. 2 Registration procedure

2. After receiving $< S_3, V_1^*, C_2, N, T_{LA1} >$ message, the cloud server selects its random number $r_{PCS} \epsilon Z_q^*$ computes $K_{PCS1} = h(ID_u \parallel S_3) \bigoplus h(r_{PCS} \parallel S_3)$ verify $T_{LA2} - T_{LA1} \leq \lhd T$, if found within the pre-defined time interval, continue onward, else, terminate it. Further, compute: $S_3^* = r_{PCS} \bigoplus h(ID_u \parallel V_1), V_2^* = h(ID_u \parallel V_1 \parallel r_{PCS}) \bigoplus MSK_{PCS}$, verifies $V_2^*? = V_2$, if validated, proceed; else, terminate. Computes $N^* = h(ID_u \parallel V_1^* \parallel C_2 \parallel V_2^* \parallel T_{LA1})$ and again checks $N^*? = N, V_1^{**} = h(ID_u^* \parallel K_{PCS}) \bigoplus h(S_3 \parallel r_{PCS})$ selects another random number $r_{PCS1} \epsilon Z_q^*$, timestamp T_{LA3} and compute $K_{PCS}^* = h(V_1^{**} \parallel C_2 \parallel V_2^* \parallel N^* \parallel T_{LA3}), SK_{PCS} = h(ID_u^* \parallel ID_{PCS} \parallel K_{PCS}^* \parallel N^*.g \parallel r_{u1}.r_{pcs1}.g \parallel V_2^* \parallel T_{LA3}), W = h(r_{u1}.g \parallel r_{PCS1}.g \parallel r_{u1}.r_{pcs1}.g \parallel T_{LA3}), X = SK_{PCS} \bigoplus h(ID_u^* \parallel V_2 \parallel ID_{PCS}), Y = ID_{PCS} \bigoplus h(S_3 \parallel r_u \parallel N^* \parallel V_1^* \parallel V_2^{**})$ and send back $< W, X, Y >$ message towards user over a public channel.
3. After Collecting $< W, X, Y >$ message from the cloud server, the user first checks the time validity $T_{LA} - T_{LA4} \leq \lhd T$ if it is true, then proceeds further and computes $K_{u2} = h(V_1^* \parallel C_2 \parallel V_2 \parallel N \parallel T_{LA3}), W^* = h(r_{u1}.g \parallel r_{PCS1}.g \parallel r_{u1}.r_{pcs1}.g \parallel T_{LA3})$, verifies $W^*? = W, X^* = SK_{PCS} \bigoplus h(ID_u \parallel V_2 \parallel ID_{PCS}^*), Y^* = ID_{PCS} \bigoplus h(S_3 \parallel r_u \parallel N^* \parallel V_1^* \parallel V_2^{**})$ and $SK_u = h(ID_u^* \parallel ID_{PCS} \parallel K_{PCS}^* \parallel N^*.g \parallel r_{u1}.r_{pcs1}.g \parallel V_2^* \parallel T_{LA3})$ and establish it the secret session key as shown in Fig. 3.

Password change procedure

1. Input $ID_u^{'}$ and $PW_u^{'}$.
2. After received $ID_u^{'}$ and $PW_u^{'}$, computes $r_u^* = S_1 \bigoplus h(ID_u^{'} \parallel PW_u^{'}), PID_u^{'} = h(ID_u^{'} \parallel r_u^* \parallel PW_u^{'}), V_2^* = V_3 \bigoplus PID_u^{'}$, and verifies $S_2 =^? h(PW_u^{'} \parallel S_1)$ if it is true, then permit to change the password; otherwise, terminate the request. $r_u^{NEW} = S_1 \bigoplus h(ID_u^{'} \parallel PW_u^{NEW}), PID_u^{NEW} = h(ID_u^{'} \parallel r_u^{NEW} \parallel PW_u^{NEW}), S_1^{NEW} = r_u^{NEW} \bigoplus h(ID_u^{'} \parallel PID_u^{NEW}), V_3^{NEW} = V_2 \bigoplus PID_u^{NEW}$.
3. After computing the value, replace $\{PW_u, S_1, S_3\}$ with $\{PW_u^{NEW}, S_1^{NEW}, S_3^{NEW}\}$ and update the password.

Security analysis

In this section, we scrutinize the security of the proposed authentication protocol, both formally using BAN logic, Proverif2.03, and informally using a pragmatic illustration. The detailed security proof is as under:

Formal security analysis

Here we have shown how the protocol provides trust and, freshness, correctness and how the protocol develops attacks based on a lack of security features. It also discusses why broad authentication protocol assaults occur and how to deal with them on the basis of trustworthiness and freshness. Keeping all of these problems in mind, we formally examine them using the two methodologies that researchers commonly employ.

ProVerif 2.03 simulation

ProVerif 2.03 is the widely used verification software toolkit for checking session key reachability, confidentiality, and secrecy. Upon simulating, we first define two channels (secure and public) and then define events,

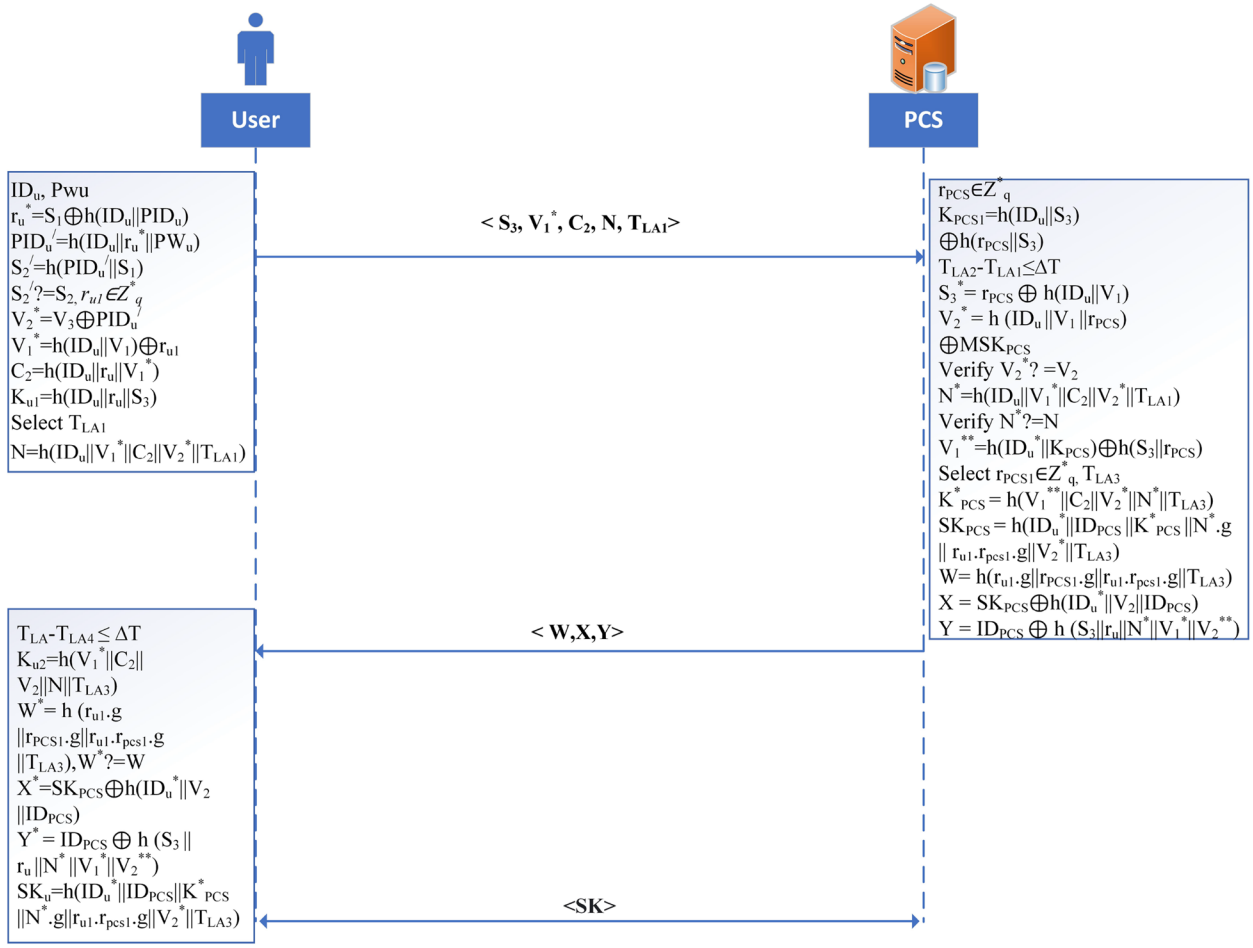

Fig. 3 Key-agreement procedure

constraints, and functions. The result generated demonstrates that the secret session key (S_k), at any stage, could not be attacked by an attacker, as given in Fig. 4.

BAN logic

A logic of trust, belief, and correctness of the protocol was first presented by Bahrower-Abdi-Nedhem in 1990 [54] and named BAN logic. Table 3 shows the primary symbols of BAN logic. The correctness of the proposed protocol is verified using the BAN (Burrows-Abadi-Needham) logic of belief.

BAN Logic Rules: The following basic BAN logic rules are well-defined including:

Message Meaning:

$$\frac{u \mid\equiv u \overset{S_K}{\longleftrightarrow} PCS, \lhd \{X\}}{u \mid\equiv PCS \mid\sim X} \tag{1}$$

If u believe that u and PCS share S_K and sees message X, then u believe PCS once said X.

Message Meaning:

$$\frac{u \mid\equiv \#(X), PCS \mid\sim X}{u \mid\equiv PCS \mid\equiv X} \tag{2}$$

If u believes that message X is fresh and that PCS once said X, then u believes that PCS believes message X.

Jurisdiction Rule:

$$\frac{u \mid\equiv PCS \mid\Rightarrow (X), u \mid\equiv PCS \mid\equiv u}{PCS \mid\equiv u} \tag{3}$$

If u believes PCS control X because it has under the jurisdiction of both peer and u believes that PCS believes message X, then u believes message X.

Session Key Rule:

$$\frac{u \mid\equiv \#(X), u \mid\equiv PCS \mid\equiv X!}{u \mid\equiv X \overset{S_K}{\longleftrightarrow} PCS} \tag{4}$$

```
(*=======CHANNELS=======*)
fre Secch:channel [private].
fre Pubch:channel.
(*=======VARIABLES=======*)
fre SKpcs:bitstr [private].
fre PWu:bitstr [private].
fre KRQ:bitstr [pravate].
fre u:bitstring [private].
fre PIDu:bitstring.
(*=======CONSTANTS=======*)
constt IDu:bitstr [private].
constt ru:bitstr [private].
constt ru1 bitstring.
constt rpcs bitstring [private].
constt rpcs1 bitstring.
(*=======FUNCTIONS=======*)
fun con(bitstr,bitstr):bitstr.
fun xor(bitstr,bitstr):bitstr.
fun h(bitstr):bitstr.
(*=======EVENTS=======*)
eventt U(bitstr).
eventt PCS(bitstr).
(*=======QUERIES=======*)
querry attacker(KCH).
querry attacker(KRQ).
querry id:bitstr;inj-eventt(U(id))==>inj-
eventt(U(id)).
querry id:bitstr;inj-eventt(PCS(id)==>inj-
eventt(PCS(id)).
```

```
(*========== REGISTRATION==========*)
let UReg=
out(Secch,(IDu));
in(Secch, (IDu: bitstring,
PIDu:bitstring));
let PIDu=h(con(con(IDu, ru1), PWu)) in
in(Secch,(V1: bitstring, V3: bitstring,
G:bitstring, g:bitstring));
let S1=xor(ru, h(con(IDu, PIDu))) in
let S2=h(con(PIDu, S1)) in
let S3=xor(ru, h(xor(IDu, V1))) in
0
).
let PCSReg=
out(Secch,(IDpcs));
in(Secch, (IDu: bitstring,
PIDu:bitstring));
let V1=xor(IDu, rpcs) in
let V2=h(con(con(IDu, SKpcs), rpcs)) in
let V3=xor(V2, PIDu) in
out(Secch, (V1: bitstring, V3: bitstring,
G:bitstring, g:bitstring));
0
).
let (!UReg)|(!PCSReg))
```

```
(*====== LOGIN & AUTHENTICATION=======*)
Let U=
out(Pubch,(IDu));
in(Pubch,(IDpcs1:bitstring,rpcs1:bitstirng,g:bitstring,W:bitstring
,V2strstr:bitstring));
!
(
let rustr=xor(S1(con(IDu, PIDu))) in
let PIDudash=h(con(IDudash, rustar), PWudash) in
if S2=h(con(PWudash, S1)) then
free ru1:bitstring.
let V2str=xor(V3, PIDudash) in
let V1str=xor(h(IDu, V1), ru1) in
let C2=h(con(IDu, ru), V1str) in
let Ku1=h(con(IDu, ru), S3) in
free TLA1:bitstring.
in(Pubch,(S3:bitstring,IDu:bitstirng,v1str:bitstring,C2:bitstring,
ru1:bitstring,g:bitstring,N:bitstring,TLA1:bitstring));
let Ku2=h(con(con(con(con(V1strstr, C2), V2), N), TLA3)) in
let V2strtstr=xor(V2strstrdash, h(con(con(V2str, IDu), IDpcsstr))
in
if W=h(con(con(con(con(con(con(rua, g), ru1), g),rpcs1), g),
TLA1)) in
let IDpcsstr=xor(IDpcs1, xor(h(S2, ru)), con(con(N, V1str),
V2strtstr)) in
let Sku=h(con(con(con(con(con(con(con(IDu, IDpcsstr), Ku2), Ng),
ru1), V2strtstr), TLA3)) in
0
).
let PCS=
out(Pubch, IDpcs);
in(Pubch,
(S3:bitstring,IDu:bitstirng,v1str:bitstring,C2:bitstring,ru1:bitst
ring,g:bitstring,N:bitstring,TLA1:bitstring));
!
(
in(0ch1,
(IDpcs1:bitstring,rpcs1:bitstirng,g:bitstring,W:bitstring,V2strstr
:bitstring));
let S3str=xor(rpcs, h(xor(IDu, V1))) in
if S3str=S3 then
free TLA1:bitstring.
let V2=h(con(con(con(IDu, SKpcs), V1), rpcs)) in
let Nstr=h(con(con(con(con(IDu, V1strstr), C2str), V2), TLA1)) in
if Nstr=N then
free TLA3:bitstring.
let V1strtstr=h(con(IDustr, SKpcs), xor(S3, rpcs), con(rpcs,
TLA1)) in
free rpcs1:bitstring.
let Kpcs1=h(con(con(con(con(V1strstr, C2str), V2), Nstr), TLA3))
in
let SKpcs=h(conc(con(con(con(con(con(IDustr, IDpcs), Kpcs1),
Nstr), rpcs1), V2strstr), TLA3)) in
Let W=h(con(con(ru1, rpcs1), TLA3)) in
let V2strstrdash=xor(V2strstr, h(con(IDustr, IPpcs)) in
let IDpcs1=xor(IDpcs, h(con(con(con(rpcs, Nstr), V2), V2strstr))
in
0
).
                    proces ((! u) | (! PCS))
```

```
SIMULATION RESULT
Compleeting equaations...
Compleeting equaations...
---Proces 1---Querry not attacker(Sk[]) in process 1
Completing equations...
Completing equations...
Starting query, not attacker(Sk[])
RESULT, not attacker (Sk[]) is true.
RESULT inj-event(end_u(id)) ==> inj-event(start_u(id)) is true.
RESULT inj-event(end_PCS(id)) ==> inj-event(start_PCS(id)) is
true.
----------------------------------------
Verification summary
Querry, not attacker (Sku[]) is tru.
Querry inj-eventt(end_u(IDu[])) ==> inj-eventt(start_u(IDu[])) is
true.
Querry inj-eventt(end_PCS(IDpcs[])) ==> inj-
eventt(start_PCS(IDpcs[])) is tru.
----------------------------------------
```

Fig. 4 ProVerif code and result

If u believes that message X is fresh, and PCS believes X, then u believes that they shared session key

Goals:

$$Goal_1 = u \ |\equiv \ (u \xleftrightarrow{S_K} PCS) \tag{5}$$

Table 3 BAN logic basic symbols

Symbol	Pronounced	Descriptions
$\|\sim$	Once Said	*P* $\|\sim$ *Q, means P Once Said Q*
$\|\equiv$	Belief	*P* $\|\equiv$*, means P believes Q*
$\lhd$	Sees	*P* $\lhd$*, means P Sees Q*
#	Fresh	Freshness rule
$\|\Rightarrow$	Jurisdiction	Jurisdiction rule: Who knows whom
$< . >$	Combines	Concatenation of two parameters
$\xleftrightarrow{K}$	Communicates	Shared key rule

$$Goal_2 = u \ |\equiv \ PCS \ |\equiv \ (u \xleftrightarrow{S_K} PCS) \tag{6}$$

$$Goal_3 = PCS \ |\equiv \ (u \xleftrightarrow{S_K} PCS) \tag{7}$$

$$Goal_4 = PCS \ |\equiv \ u \ |\equiv \ (u \xleftrightarrow{S_K} PCS) \tag{8}$$

Idealize Form:

$$Message_1.u \longrightarrow PCS : \{S_3, V_1^*, C_2, N, T_{LA1}\} \tag{9}$$

$$Message_2.PCS \longrightarrow u : \{W, X, Y\} \tag{10}$$

Initial Assumptions:

$$A_1.u \ |\equiv \#(r_{u1}) \tag{11}$$

$$A_2.u \ |\equiv \#(r_u^*) \tag{12}$$

$$A_3.u \mid\equiv PCS \xleftrightarrow{SK_{PCS}} u) \tag{13}$$

$$A_4.u \mid\equiv u \xleftrightarrow{ID_u, PID_u} PCS) \tag{14}$$

$$A_5.PCS \mid\equiv \#(r_{pcs}) \tag{15}$$

$$A_6.PCS \mid\equiv PCS \xleftrightarrow{SK_{PCS}} u) \tag{16}$$

$$A_7.PCS \mid\equiv PCS \xleftrightarrow{V_1, V_3, G, g, h(.)} u) \tag{17}$$

Keeping the assumptions mentioned above in mind, we proceed step by step to achieve the aforementioned objectives. So, according to $Message_1$ and A_1, we get,

$$A_8.u \mid\equiv r_{u1} \longrightarrow \{S_3, V_1^*, C_2, N, T_{LA1}\} \tag{18}$$

By taking A_3 and A_8, we get,

$$A_9.u \mid\equiv \{S_3, V_1^*, C_2, N, T_{LA1}\} \tag{19}$$

Now, taking A_2 and A_9, we get,

$$A_{10}.u \mid\equiv \#(r_u^*) \rightarrow \{S_3, V_1^*, C_2, N, T_{LA1}\} \tag{20}$$

Finally, by taking A_3 and A_{10}, we get,

$$A_{11}.u \mid\equiv\rightarrow \{S_3, V_1^*, C_2, N, T_{LA1}\} \tag{21}$$

A_{11} can also be written as:

$$A_{12}.u \mid\equiv (u \rightarrow \{S_3, V_1^*, C_2, N, T_{LA1})\} \tag{22}$$

The A_{12} can be written as:

$$A_{13}.u \mid\equiv (u \xleftrightarrow{S_K} \{S_3, V_1^*, C_2, N, T_{LA1})\} \tag{23}$$

or

$$A_{14}.u \mid\equiv (u \xleftrightarrow{S_K} PCS)\} \tag{24}$$

Getting A_{14} we achieved **Goal1**.

Taking $message_1$ and A_5, we get,

$$A_{15}.u \mid\equiv PCS \rightarrow \#(r_{PCS}) : \{S_3, V_1^*, C_2, N, T_{LA1}\} \tag{25}$$

Take A_4 and A_{15}, and we get

$$A_{16}.u \rightarrow PCS \mid\equiv u \rightarrow (u \xleftrightarrow{S_K} PCS), \tag{26}$$

We can also write this equation as:

$$A_{17}.u \mid\equiv PCS \mid\equiv (u \xleftrightarrow{S_K} PCS) \tag{27}$$

Getting A_{17} we achieved **Goal2**.

For the following two goals, let's take $Message_2$ and A_6

$$A_{18}.PCS \rightarrow\mid\equiv PCS \xleftrightarrow{SK_{PCS}} u : \{W.X, Y\} \tag{28}$$

A_7 and A_{18} becomes

$$A_{19}.PCS \rightarrow\mid\equiv \xleftrightarrow{V_1, V_3, G, g, h(.)} u : \{W.X, Y\} \tag{29}$$

$$A_{20}.PCS \xleftrightarrow{V_1, V_3, G, g, h(.)} u :\mid\equiv \{W.X, Y\} \tag{30}$$

A_{18} and A_{20} can write as

$$A_{21}.PCS \xleftrightarrow{SK_{PCS}} u \mid\equiv \{W.X, Y\} \tag{31}$$

$$A_{22}.PCS \mid\equiv u \xleftrightarrow{SK_{PCS}} \{W.X, Y\} \tag{32}$$

$$A_{23}.PCS \mid\equiv (u \xleftrightarrow{SK_{PCS}} PCS) \tag{33}$$

Getting A_{23} we achieved **Goal3**.

Now, for the $Goal_4$, we will take A_{23} and A_2.

$$A_{24}.PCS \mid\equiv (u \xleftrightarrow{SK_{PCS}} PCS) : \#(r_u^*) \tag{34}$$

Taking A_{21} and A_{24}, we get

$$A_{25}.PCS \xleftrightarrow{SK_{PCS}} u :\mid\equiv \xleftrightarrow{SK_{PCS}} PCS) : \#(r_u^*) \tag{35}$$

Take A_{19} and A_{25}, and we get

$$A_{26}.PCS \mid\equiv u :\mid\equiv \#(r_u^*) \tag{36}$$

or

$$A_{27}.PCS \mid\equiv u \mid\equiv (u \xleftrightarrow{SK_{PCS}} PCS) \tag{37}$$

From A_{26} and A_{27}, we get that

$$A_{28}.PCS \mid\equiv u \mid\equiv (u \xleftrightarrow{S_K} PCS) \tag{38}$$

Getting A_{28} we achieved **Goal4**.

Informal security analysis

We present a pragmatic discussion of different attacks for achieving the key security and feature attributes. These are as follows:

1. Resists Offline Password Guessing Attack: Let's suppose the adversary $\forall$ guesses the $ID_\forall, PW_\forall$, and random number $r_\forall$ and computes $PID_\forall = h(ID_\forall \| r_\forall \| PW_\forall)$ and later launches the attack. After checking ID_u, PID_u, and verifying the condition $S_2 =^? h(PID_\forall \| S1)$. The adversary $\forall$ tests $ID_\forall =^? IDu, PW_\forall =^? PWu$, and $PID_\forall =^? PIDu$. However, $ID_\forall$ and $PW_\forall$ are finite values, while $r_\forall$ is

too hard to guess; thus, our scheme resists offline password-guessing attacks.

2. Free from De-synchronization Attack: There is no need to update the proposed scheme's parameters on the user or *PCS* side. In contrast, in the case of the password update phase, the entities validate each other. Therefore, the user and *PCS* do not require synchronization properties in the proposed scheme.
3. Provision of Key Agreement: In our proposed protocol, the user and *PCS* authenticate each other with $ID_u^* = ID_u, ID_{PCS}^* = ID_{PCS}, \quad N^* = N, V_2* =^? V_2$ and agree on $S_K = S_{Ku} = S_{KPCS}$.
4. Spoofing Attack: SK_{PCS} is a secret key of *PCS*; hence ∀ cannot calculate $V_2 = h(ID_u \| V_1 \| r_{PCS})$ and $N^* = h(ID_u \| V_1^* \| C_2^* \| V_2^* \| T_{LA1})$ Thus $N^* \neq N$ Therefore, ∀ cannot launch a spoofing attack.
5. Resists Insider Attack: In the proposed protocol registration phase u calculate $PID_u = h(ID_u \| r_u 1 \| PW_u)$, where r_{u1} is a random number, and thus the administrators are unable to get PID_u. Therefore, our proposed scheme resists insider attacks.
6. Perfect Message Authentication: The *PCS* received $Message_1 = (S_3, V_1^*, C_2, N, T_{LA1})$ and verified $T_{LA2} - T_{LA1} \leqslant \triangle T, V_2^* =^? V_2$ and $N^* =^? N$ then sent $Message_2 = (W, X, Y)$ toward the u, and the user verified $T_{LA4} - T_{LA3} \leqslant \triangle T$, $W \stackrel{?}{=} W, X =^? X$ and $Y =^? Y$. After the verification process, the message will not be accepted if any message fails. Therefore, authentication is verified between u and PCS
7. Support Anonymity: In the proposed protocol, the PCS uses the anonymous identity of u ID_u^* and for itself ID_{PCS}^*. Therefore, the identity is untraceable.
8. Offers Mutual Authentication: In the proposed protocol login and authentication phase, the *PCS* compute $W = h(r_{u1}.g \| r_{PCS1}.g \| r_{u1}.r_{pcs1}.g \| T_{LA3})$ and u verify $W \stackrel{?}{=} h(r_{u1}.g \| r_{PCS1}.g \| r_{u1}.r_{pcs1}.g \| T_{LA3})$ thus, u and *PCS* mutual authenticate each other.
9. Resists Replay Attack: The proposed protocol uses timestamps and random numbers to resist replay attacks. LAP timestamp $T - T \leqslant \triangle T$, where $\triangle T$ is the max time delay. Moreover, u and *PCS* used different random numbers. If the ∀ tries to launch a replay attack, the proposed scheme verifies $W \stackrel{?}{=} h(r_{u1}.g \| r_{PCS1}.g \| r_{u1}.r_{pcs1}.g \| T_{LA3}), N^* = N$ and session key $S_K = S_{Ku} = S_{KPCS}$ Thus, the proposed scheme is secure against reply attacks.
10. Resists Stolen Verifier Attack: Our proposed scheme does not store password verification and validation tables in the database on the *PCS* side. Thus, ∀ cannot masquerade as a u to mislead the *PCS* in the authentication process. Therefore, our scheme resists stolen verifier attacks.
11. Free of Impersonation Attack: The ∀ commonly use two techniques in order to impersonate u. The first one ID_u, and the second one is PW_u, and it uses these two techniques, the ∀ constructs $Messgae_1 = (S_3, V_1^*, C_2, N, T_{LA1})$. However, the stolen verifier and offline password guessing attack are impossible for ∀. Thus, our scheme is more secure against user impersonation attacks.
12. Man-in-The-Middle (MITM) Attacks: In our proposed protocol, the user (u) and next-generation public cloud server (*PCS*) share session keys after the mutual authentication. The ∀ needs u, ID_u, PW_u, and *PCS* secret key S_{KPCS} to pass verification. The ∀ also cannot guess random num-

Table 4 Security features comparison

Schemes → Features ↓	[45]	[55]	[37]	[56]	[54]	[70]	[71]	[72]	[69]	[7]	Our
SG1	√	×	√	×	√	√	√	√	×	×	√
SG2	√	√	√	×	×	×	√	×	×	×	√
SG3	×	×	√	×	√	√	√	√	×	×	√
SG4	√	×	√	√	×	×	×	×	√	√	√
SG5	√	√	√	√	√	√	√	√	√	√	√
SG6	×	×	×	×	×	√	×	√	×	×	√
SG7	√	×	√	√	√	√	√	√	√	√	√
SG8	×	×	√	×	√	√	√	√	×	×	√
SG9	√	√	√	√	√	√	√	√	√	√	√
SG10	√	√	√	√	√	√	√	×	√	√	√
SG11	√	×	√	√	√	√	×	√	√	√	√

bers r_u, r_{PCS}. Therefore, the ∀ cannot construct a connection with *PCS* and *u*; thus, the proposed scheme resists man-in-the-middle attacks.

Performance analysis

This section is carried out by considering four features of protocol performance, including the security comparison shown in Table 4, storage overheads, communication costs, and computation time complexity; each of these can be described for the proposed protocol as under:

Security feature

We have compared our proposed protocol to existing schemes in this part of the article, such as [7, 23, 31, 40–42, 69724 shows that our proposed scheme resists all known possible security threats.

Storage overheads

First of all, we considered the work done by Kilinc et al. [73]. The hash code is 160 bits, the identity is 64 bits, the password is 60 bits, and the timestamp is 56 bits. Therefore, keeping this in view, the storage process for the proposed protocol is considered only in the registration phase, i.e., $S_1 = 320$ bits, $S_2 = 160$ bits, $S_3 = 320$ bits, $g = 160$ bits, $G = 160$ bits, $h(.) = 160$ bits, $ID = 64$ bits, $PW = 60$ bits, and timestamp = 56 bits. $(320 + 160 + 320 + 160 + 160 + 160 + 64 + 60 + 56 = 1460)$.

Communication cost

In this section, our proposed scheme communication cost is discussed and calculated. The communication cost is calculated using the messages transmitted between the *u* and the *PCS* in the login and authentication. *u* to $PCSMessgae_1 = (S_3, V_1^*, C_2, N, T_{LA1}), (160 + 160) + (160 + 160) + (160) + (160) + (56) = 1016$, while from *PCS* to $_{uMessage_2 = (W,X,Y)}$, $(160 + 56) + (160 + 160) + (64 + 160) = 760$. Therefore, the total communication cost of our proposed protocol is 1776 bits. Moreover, the communication cost comparison of our proposed scheme with [23, 31, 40–42, 69–72] is shown in Fig. 5. Figure 5 shows that our scheme is lightweight than the existing schemes, except [69]. However, the computation cost of [69] is much higher than our scheme.

Computation cost

In this section, we consider the work accomplished by [73] that used a smartphone with a memory of 4 GB, a CPU size of an octa-core 2.01 GHz processor, and a laptop with an intel core $i7 - 2620M(2.7GHzprocessor)$, and 4 GB of RAM. Keep in mind that the first is a user and the last is a next-generation public cloud server. Let's suppose the computation cost for a one-way hash function is denoted by T_h,XOR, T_ϕ, ECC point of multiplication T_M, point of an addition T_A, random number extraction T_R. Figure 6 shows that our scheme achieved better computation costs than existing schemes. Furthermore, the computation comparison of our scheme with the existing protocols is illustrated in Fig. 6.

Conclusion

Nowadays, users outsource a tremendous amount of data remotely to the next-generation public cloud. The main concern with data broadcasting is how to

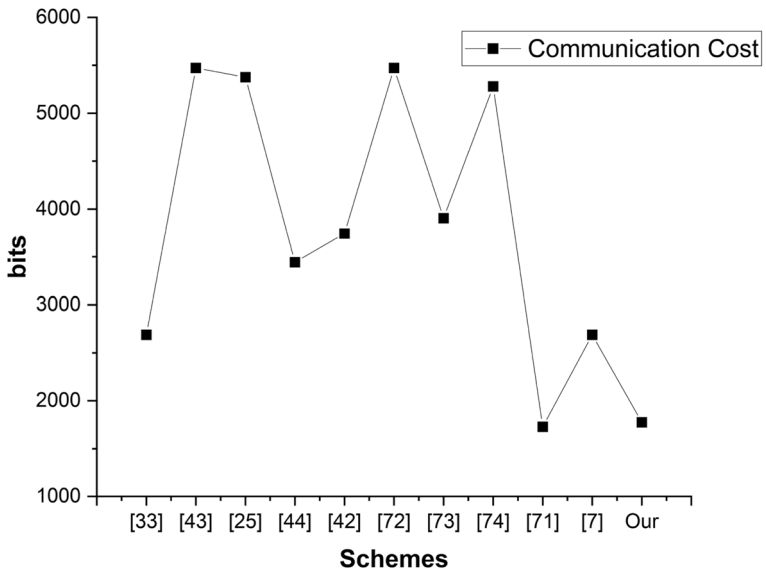

Fig. 5 Communication cost comparison

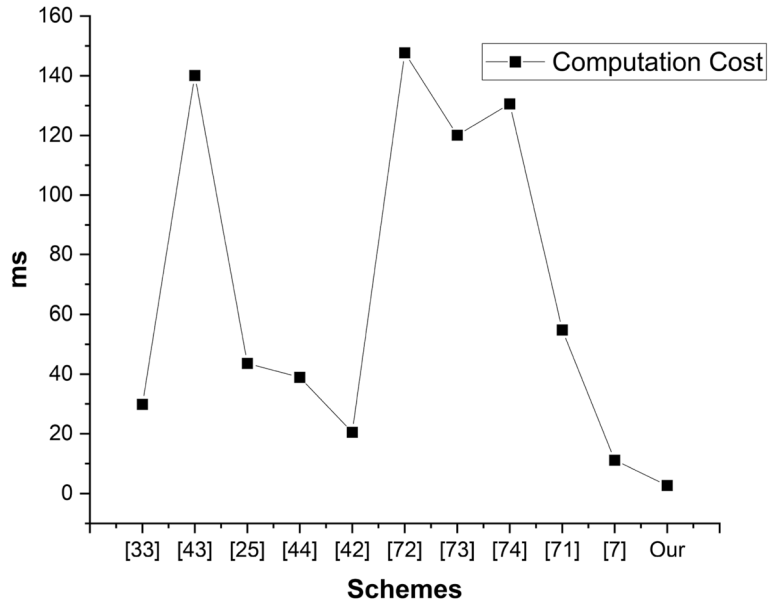

Fig. 6 Computation cost comparison

securely access it, which is a big challenge for researchers. Therefore, we attempted to make it secure for both parties. In this regard, we have chosen a lightweight public cryptographic technique such as ECC and, after checking security using BAN logic and ProVerif2.03 simulation. As a result, it concludes that the scheme is verifiably secure against all known attacks. Furthermore, the performance analysis section evaluated the key features. The result shows that the proposed scenario is lightweight, efficient, effective, practical, and recommended for next-generation public cloud computing. We intend to improve our proposed authentication process for end users in the future to protect against quantum attacks. Simultaneously, its security will be tested using the AVISPA simulator.

Authors' contributions

Naveed Khan, Zhang Jianbiao, and Shehzad Ashraf Chaudhry proposed the first draft. Naveed Khan, Jehad Ali, and Huhnkuk Lim wrote the literature section. The security analysis section is handled by Naveed Khan, Shehzad Ashraf Chaudhry, Intikhab Ullah and Huhnkuk Lim. Naveed Khan, Muhammad Salman Pathan and Shehzad Ashraf Chaudhry complete the performance analysis section. Naveed Khan, Zhang Jianbiao, and Shehzad Ashraf Chaudhry approve the final draft.

Funding

This work was supported by the Korean Government (Ministry of Science and ICT) through the National Research Foundation of Korea (NRF) under Grant 2021R1A2C1010481.

Availability of data and materials

The first author will provide the supporting data for this work upon reasonable request.

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 3 January 2023 Accepted: 26 May 2023
Published online: 08 July 2023

References

1. Chaudhry SA (2021) Combating identity de-synchronization: an improved lightweight symmetric key based authentication scheme for IoV. J Netw Intell 6:12
2. Jadeja Y, Modi K (2012) Cloud computing-concepts, architecture and challenges. pp 877-880
3. Liu F, Tong J, Mao J, Bohn R, Messina J, Badger L, Leaf D (2011) NIST cloud computing reference architecture. NIST Spec Publ 500(2011):1–28
4. Parkhill DF (1966) Challenge of the computer utility. Addison-Wesley.
5. Thakkar N, Vaghela R (2018) Secure Model for Session Hijacking using Hashing Algorithm. Int J Adv Res Innov Ideas Educ 4(3):70–75
6. Khan N, Zhang J, Ali J, Pathan MS, Chaudhry SA (2022) A Provable Secure Cross-Verification Scheme for IoT Using Public Cloud Computing. Secur Commun Netw 2022;11. Article ID 7836461. https://doi.org/10.1155/2022/7836461.
7. Chaudhry SA (2021) Correcting PALK: Password-based anonymous lightweight key agreement framework for smart grid. Int J Electr Power Energy Syst 125:106529
8. Chunka C, Banerjee S, Nag S, Goswami RS (2021) A Secure Key Agreement Protocol Defiant to Denial-of-Service Attack based on Three Party Authentication. J Inst Eng India B 103;1-12
9. Chaudhry SA, Naqvi H, Mahmood K, Ahmad HF, Khan MK (2017) An improved remote user authentication scheme using elliptic curve cryptography. Wirel Pers Commun 96(4):5355–5373

10. Khan AA, Kumar V, Ahmad M (2010) An elliptic curve cryptography based mutual authentication scheme for smart grid communications using biometric approach. J King Saud Univ-Comput Inform Sci 34(3):698–705
11. Dolev D, Yao A (1983) On the security of public key protocols. IEEE Trans Inf Theory 29(2):198–208
12. Chaudhry SA, Nebhen J, Yahya K, Al-Turjman F (2021) A privacy enhanced authentication scheme for securing smart grid infrastructure. IEEE Trans Ind Inform 18:5000–5006
13. Canetti R, Goldreich O, Halevi S (2004) The random oracle methodology, revisited. J ACM (JACM) 51(4):557–594
14. Motta G, Sfondrini N, Sacco D (2012) Cloud computing: An architectural and technological overview. 2012 International Joint Conference on Service Sciences. IEEE. pp 23-27
15. Fu J-S, Liu Y, Chao H-C, Bhargava BK, Zhang Z-J (2018) Secure data storage and searching for industrial IoT by integrating fog computing and cloud computing. IEEE Trans Ind Inform 14(10):4519–4528
16. Ali A, Zhu Y, Zakarya M (2021) A data aggregation based approach to exploit dynamic spatio-temporal correlations for citywide crowd flows prediction in fog computing. Multimedia Tools Appl 1-33
17. Khan N, Zhang J, Jan SU (2022) A Robust and Privacy-Preserving Anonymous User Authentication Scheme for Public Cloud Server. Secur Commun Networks 2022
18. Lamport L (1981) Password authentication with insecure communication. Commun ACM 24(11):770–772
19. Hwang T, Chen Y, Laih CJ (1990) Non-interactive password authentications without password tables. pp 429-431
20. Jiang P, Wen Q, Li W, Jin Z, Zhang H (2015) An anonymous and efficient remote biometrics user authentication scheme in a multi server environment. Front Comput Sci 9:142–156
21. Lin H, Wen F, Du C (2015) An improved anonymous multi-server authenticated key agreement scheme using smart cards and biometrics. Wirel Pers Commun 84(4):2351–2362
22. He D, Wang D (2014) Robust biometrics-based authentication scheme for multiserver environment. IEEE Syst J 9(3):816–823
23. Odelu V, Das AK, Goswami A (2015) A secure biometrics-based multi-server authentication protocol using smart cards. IEEE Trans Inf Forensic Secur 10(9):1953–1966
24. Amin R, Biswas G (2015) Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment. Wirel Pers Commun 84(1):439–462
25. Hsieh W-B, Leu J-S (2014) An anonymous mobile user authentication protocol using self-certified public keys based on multi-server architectures. J Supercomput 70(1):133–148
26. Chandrakar P, Om H (2017) A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC. Comput Commun 110:26–34
27. Park Y, Park Y (2016) Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. Sensors 16(12):2123
28. Choi Y, Lee Y, Won D (2016) Security improvement on biometric based authentication scheme for wireless sensor networks using fuzzy extraction. Int J Distrib Sensor Netw 12(1):8572410
29. Yoon E-J, Kim C (2013) Advanced biometric-based user authentication scheme for wireless sensor networks. Sens Lett 11(9):1836–1843
30. Irshad A, Sher M, Chaudhary SA, Naqvi H, Farash MS (2016) An efficient and anonymous multi-server authenticated key agreement based on chaotic map without engaging Registration Centre. J Supercomput 72(4):1623–1644
31. Reddy AG, Yoon E-J, Das AK, Odelu V, Yoo K-Y (2017) Design of mutually authenticated key agreement protocol resistant to impersonation attacks for multi-server environment. IEEE Access 5:3622–3639
32. Chuang Y-H, Lei C-L, Shiu H-J (2020) Cryptanalysis of Four Biometric Based Authentication Schemes with Privacy-preserving for Multi-server Environment and Design Guidelines. pp 66-73
33. Dharminder D, Mishra D, Li X (2020) Construction of RSA-Based Authentication Scheme in Authorized Access to Healthcare Services: Authorized Access to Healthcare Services. J Med Syst 44:1–9
34. Lin C-H, Tien C-W, Pao H-K (2012) Efficient and effective NIDS for cloud virtualization environment. pp 249-254
35. Wu H, Ding Y, Winer C, Yao L (2010) Network security for virtual machine in cloud computing. 5th International conference on computer sciences and convergence information technology. pp 18-21
36. Du P, Nakao A (2010) DDoS defense as a network service. 2010 IEEE Network Operations and Management Symposium-NOMS 2010. pp 894-897
37. Priyadarshini R, Barik RK (2022) A deep learning based intelligent framework to mitigate DDoS attack in fog environment. J King Saud Univ-Comput Inf Sci 34(3):825-831
38. Krishnan D, Chatterjee M (2012) An adaptive distributed intrusion detection system for cloud computing framework. Recent Trends in Computer Networks and Distributed Systems Security: International Conference, SNDS 2012, Trivandrum, India, October 11-12, 2012. Proceedings 1. pp 466-473
39. Islam S, Ouedraogo M, Kalloniatis C, Mouratidis H, Gritzalis S (2015) Assurance of security and privacy requirements for cloud deployment models. IEEE Trans Cloud Comput 6(2):387–400
40. Li W, Xuelian L, Gao J, Wang HY (2019) Design of secure authenticated key management protocol for cloud computing environments. IEEE Trans Dependable Secure Comput
41. Liao Y-P, Hsiao C-M (2013) A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients. Futur Gener Comput Syst 29(3):886–900
42. He D, Zeadally S, Kumar N, Wu W (2016) Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. IEEE Trans Inf Forensic Secur 11(9):2052–2064
43. Pramanik S, Sakkari D, Pramanik S (2022) Privacy conserving authenticated key settlement approach for remote users in IoT based Telecare Medicine information system. Smart Health 26:100355
44. Wang C, Li S, Ma M, Tong X, Zhang Y, Zhang B (2022) A Noval and Efficient ECC-Based Authenticated Key Agreement Scheme for Smart Metering in the Smart Grid. Electronics 11(20):3398
45. Nyangaresi VO (2022) Lightweight Anonymous Authentication Protocol for Resource-Constrained Smart Home Devices Based on Elliptic Curve Cryptography. J Syst Archit 133:102763
46. Qiu S, Xu G, Ahmad H, Wang L (2017) A robust mutual authentication scheme based on elliptic curve cryptography for telecare medical information systems. IEEE Access 6:7452–7463
47. Sahoo SS, Mohanty S, Majhi B (2021) A secure three factor based authentication scheme for health care systems using IoT enabled devices. J Ambient Intell Humanized Comput 12(1):1419–1434
48. Ryu J, Oh J, Kwon D, Son S, Lee J, Park Y, Park Y (2022) Secure ECC-based three-factor mutual authentication protocol for telecare medical information system. IEEE Access 10:11511–11526
49. Xiong L, Xiong N, Wang C, Yu X, Shuai M (2019) An efficient lightweight authentication scheme with adaptive resilience of asynchronization attacks for wireless sensor networks. IEEE Trans Syst Man Cybern Syst 51:5626–5638
50. Luo H, Wen G, Su J (2020) Lightweight three factor scheme for real-time data access in wireless sensor networks. Wirel Netw 26(2):955–970
51. Roy S, Das AK, Chatterjee S, Kumar N, Chattopadhyay S, Rodrigues JJ (2018) Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications. IEEE Trans Ind Inf 15(1):457–468
52. Wang D, Li W, Wang P (2018) Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks. IEEE Trans Ind Inf 14(9):4081–4092
53. Ahmed AA, Wendy K, Kabir MN, Sadiq AS (2020) Dynamic Reciprocal Authentication Protocol for Mobile Cloud Computing. IEEE Syst J 15(1):727–737
54. Khan ZJN, Ullah I, Pathan MS, Lim H (2023) Lattice-Based Authentication Scheme to Prevent Quantum Attack in Public Cloud Environment. Comput Mater Continua 75(1):35–49
55. Liu JK, Liang K, Susilo W, Liu J, Xiang Y (2015) Two-factor data security protection mechanism for cloud storage system. IEEE Trans Comput 65(6):1992–2004
56. Ali A, Zhu Y, Zakarya M (2021) Exploiting dynamic spatio-temporal correlations for citywide traffic flow prediction using attention based neural networks. Inf Sci 577:852–870
57. Awan N, Ali A, Khan F, Zakarya M, Alturki R, Kundi M, Alshehri MD, Haleem M (2021) Modeling dynamic spatio-temporal correlations for urban traffic flows prediction. IEEE Access 9:26502–26511

58. Ali A, Zhu Y, Zakarya M (2022) Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction. Neural Netw 145:233–247
59. Ali A, Zhu Y, Chen Q, Yu J, Cai H (2019) Leveraging spatio-temporal patterns for predicting citywide traffic crowd flows using deep hybrid neural networks. pp 125-132
60. Chaudhry SA, Naqvi H, Sher M, Farash MS, Hassan MU (2017) An improved and provably secure privacy preserving authentication protocol for SIP. Peer Peer Netw Appl 10:1–15
61. Tu H, Kumar N, Chilamkurti N, Rho S (2015) An improved authentication protocol for session initiation protocol using smart card. Peer Peer Netw Appl 8(5):903–910
62. Farash MS (2016) Security analysis and enhancements of an improved authentication for session initiation protocol with provable security. Peer Peer Netw Appl 9(1):82–91
63. Varma C (2018) A study of the ECC, RSA and the diffie-hellman algorithms in network security. pp 1-4
64. Tsaur W-J (2005) Several security schemes constructed using ECC-based self-certified public key cryptosystems. Appl Math Comput 168(1):447–464
65. Farash MS (2017) An improved password-based authentication scheme for session initiation protocol using smart cards without verification table. Int J Commun Syst 30(1):e2879
66. Zhang L, Tang S, Cai Z (2014) Cryptanalysis and improvement of password-authenticated key agreement for session initiation protocol using smart cards. Secur Commun Netw 7(12):2405–2411
67. Azrour M, Farhaoui Y, Ouanan M (2018) Cryptanalysis of Farash et al.'s SIP authentication protocol. Int J Dyn Syst Differ Equat 8(1-2):77-94
68. Burrows M, Abadi M, Needham RM (1989) A logic of authentication. Proc R Soc Lond A Math Phys Sci 426(1871):233–271
69. Bera B, Chattaraj D, Das AK (2020) Designing secure blockchain-based access control scheme in IoT-enabled Internet of Drones deployment. Comput Commun 153:229–249
70. Das AK, Kumari S, Odelu V, Li X, Wu F, Huang X (2016) Provably secure user authentication and key agreement scheme for wireless sensor networks. Secur Commun Netw 9(16):3670–3687
71. Li X, Peng J, Niu J, Wu F, Liao J, Choo K-KR (2017) A robust and energy efficient authentication protocol for industrial internet of things. IEEE Internet Things J 5(3):1606–1615
72. Jiang Q, Zhang N, Ni J, Ma J, Ma X, Choo K-KR (2020) Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles. IEEE Trans Veh Technol 69(9):9390–9401
73. Kilinc HH, Yanik T (2013) A survey of SIP authentication and key agreement schemes. IEEE Commun Surv Tutor 16(2):1005–1023

Publisher's Note