Akram *et al. Hum. Cent. Comput. Inf. Sci.* (2020) 10:22
https://doi.org/10.1186/s13673-020-00227-9

Human-centric Computing and Information Sciences

RESEARCH

Open Access

An anonymous authenticated key-agreement scheme for multi-server infrastructure

Muhammad Arslan Akram[1], Zahid Ghaffar[1], Khalid Mahmood[1], Saru Kumari[2], Kadambri Agarwal[3] and Chien-Ming Chen[4*]

*Correspondence: chienmingchen@ieee.org
[4] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
Full list of author information is available at the end of the article

Abstract

Due to single-time registration, the multi-server authentication provides benefit for getting services from different servers through trusted agent. Generally, users feel hesitation for registering themselves individually with all service providers due to the problem of memorizing the multiple passwords. The multi-server authentication allows a quick access to services by real-time customer validation on public channel. Thereafter, hundreds of multi-server authentication protocols have been introduced. However, the more efficient and robust authentication schemes are being explored by the research academia. We introduce an anonymous scheme that resists the major security threats like impersonation attack, insider attack and password modification attacks in viable computing cost. We use random oracle model for formal security analysis of the proposed scheme. The performance analysis shows that the proposed scheme incurs less computation, energy, communication and storage cost as compared to related protocols. This analysis and comparison show that our proposed scheme is quite effective for the purpose of anonymous authentication and key agreement.

Keywords: Multi-server, User anonymity, Authentication, Impersonation, Key-agreement protocol, Internet of Things, Cloud computing

Introduction

Multi-server authentication (*MSA*) allows the abrupt access to various online services as compared to single registration, in peer-to-peer environment. *MSA* architecture [1, 2] is suitable for both sides, i.e. customers and service providers. Because, due to one-time registration with registration center (*RC*), the customer does not need for remembering multiple passwords. Consequently, the *MSA* architecture facilitates the service providers to maintain verifier database for each authentic user, in order to avoid multiple registration. To get perk from these services from various servers, the customers rely on a single time registration with *RC*. The *MSA* environment involves various servers (S_j), customers (U_u) and a registration center (*RC*). The generic architecture of remote-user authentication is shown in Fig. 1. The registration of each user and server with *RC* take place on secure channel. Therefore, trust flows from RC to respective users and servers.

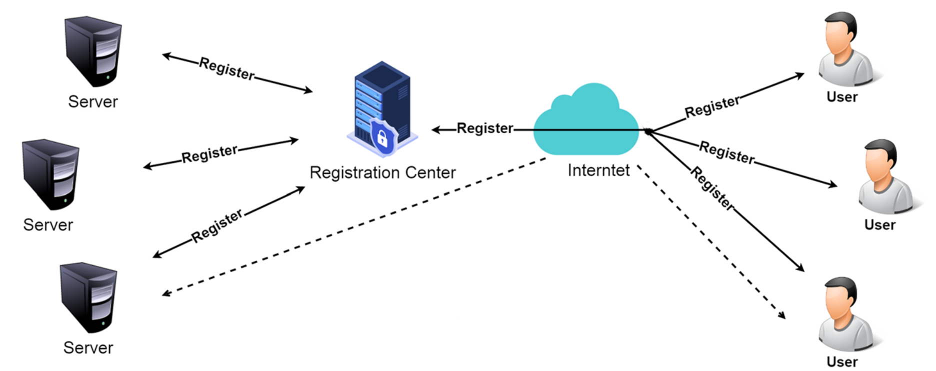

Fig. 1 Generic multi-server authentication environment

Thereafter, the customers are able to get advantages of the services provided by service providers.

Already, plenty of schemes have been introduced so far to achieve better efficiency and security [3–9]. However, it is realized on the basis of frequent security attacks that more stronger protocols need to be developed [10–14]. Initially, a key agreement protocol for *MSA* framework is presented early in 2000 by Lee and Chang [15]. Later, the protocol is found susceptible to impersonation and anonymity violation attacks [16]. After that, *MSA* scheme engaged by using *RSA* (*Rivest* − *Shamir* − *Adleman*)

Generic multi-server authentication environment

crypto-primitives and Lagrange interpolating polynomials, for a remote subscriber is presented by Tsaur [17]. The protocol [17] is compromised by password-guessing attack. Then, *MSA* protocol for a system of artificial neural network based on password is presented by Li et al. [18], which needs more time and higher cost. After that, ElGamal digital signature-dependent *MSA* protocol is presented by Lin et al. [1]. However, it is realized for smart card dependent applications that the scheme is too expensive in terms of memory requirement. Thereafter, *MSA* protocol based on symmetric crypto-system is presented by Juang [2], with an inherent scalability issue due to maintaining verifier-repository for each user at server end.

MSA protocol is then introduced by Chang and Lee [19], which is found to be vulnerable for privileged-insider and server impersonation attacks [20, 21]. Afterwards, another remote user verified key agreement protocol based on dynamic identity is presented by Liao and Wang [20] for MSA architecture. Hsiang and Shih [21] found that the protocol [20] is susceptible to spoofing and privileged insider attacks, and also introduced an enhanced protocol. Soon, Lee et al. [22] realized that the protocol [21] has no ability to attain agreement for mutual authentication and also introduced an modified scheme. However, Chen and Lee [23] observed that the protocol [24] is inadequate for smart card security and incompatible with two-factor authentication. Likewise, the protocol [24] also fails to prevent masquerading attack. Moreover, password change phase becomes more complex due to involvement of *RC*. Then, Irshad et al. [25] proves that the protocol [23] is insecure against smart card stolen attack that helps to reveal the session key and password. The protocol [23] is also vulnerable to trace and spoofing attacks. Later, due to the inefficiency and insecurity of the above mentioned schemes, many researchers have made improvements to the authentication method [26–28]. Additionally, some protocols have begun to use biometrics to ensure security [29]. The above discussion shows that designing the protocol for multi-server infrastructures to meet security requirements is a serious task. All current solutions are neither immune to all known attacks, nor they can guarantee the consumption of their own computations. Section III demonstrates our proposed scheme. Security and performance analysis are illustrated in Section IV and V, respectively. Presented work is concluded in last Section VI.

Our contribution

An anonymous three factor authentication protocol is introduced in this paper and the authentication of users with the help of biometric impression is enhanced. We encompass our contributions as follows.

1. First, we introduce an *ECC* based three-factor user authenticated key-agreement protocol.
2. Second, if smart card can be forged by an adversary, then the environment of user cannot be secure. In our introduced protocol, the verification of biometric impression of users can be done by the client as well as by the server; in some specific applications it can provide security protection for specific requirements. RC and server have separate responsibilities, as RC is involved in authentication phase. RC retains the privacy of registration and server validates the client for further service providing; it can make the protocol more scalable for multi-server the architecture.
3. Finally, our protocol offers the mutual authentication for each pair of three participants (server, user and *RC*) for providing strong protection by identifying as possible replay messages.

Preliminaries

The hash functions, elliptic curve cryptography, adversarial model which is used in this paper are stated in this section. Whereas, Table 1 is presenting the common notations, used in rest of the article.

Hash functions

By taking an input string $O = H(String)$ of random size, a fixed size output is generated by hash. Generated output is called hash code. A little change in the value of string can cause a huge difference. Whereas, a secure one way hash function has following specifications:

- If the string is described, it is easy to find $O = H(String)$.
- It is impossible to find out the string, if $O = H(String)$ is illustrated.
- It is mundane task to distinguish input of $String_1$ and $String_2$ so that $H(String_1) = H(String_2)$. This feature is called collision resistance.

Definition 1 (Characteristics of collision Resistance) Secure hash function H(.) is predetermined for collision resistance. The possibility that an attacker $\mathcal{A}$ can find a

Table 1 Common used notations

Common notations	blueElucidations
$\mathcal{U}_u$	u_{th} user of the system
$\mathcal{RC}$	Centralized registration center of the infrastructure
ID_u	Specific user's identity
PW_u	Specific user's password
B_u	Biometric identity of specific user
PID_u	User's pseudo identity
SC_u	Smart card issued to each specific user
$\mathcal{S}_j$	j_{th} service provider of the infrastructure
ID_j	Identity of service provider
x	Secret key of $\mathcal{RC}$
pk_{RC}	Public key of $\mathcal{RC}$
s	Secret key of $\mathcal{S}_j$
pk_{S_j}	Public key of $\mathcal{S}_j$
$E_p(e, f)$	An elliptic curve
P	Base point of the elliptic curve $E_p(e, f)$
$H(.)$	Function specified for Bio-hash
$h(.)$	One-way digest function of hashing
$\|\|$	Concatenation operator
$\oplus$	XoR operator

pair $(String_1 \neq String_2)$ as $H(String_1) = H(String_2)$ is separated as $Advs_{\mathcal{A}}^{HASH}(t) = Prob[(String_1, String_2) \Leftarrow_r \mathcal{A} : (String_1 \neq String_2), H(String_1) = H(String_2)]$, where attacker is allowed to select a pair $(String_1, String_2)$ randomly. Attacker's perk is calculated against the randomly selections taken up with-in polynomial time (t). The collision resistance conclude that $Advs_{\mathcal{A}}^{HASH}(t) \leq \in$, whereas $\in > 0$, is an enough tiny value.

Elliptic-curve cryptography(ECC)

The Elliptic-curve equation is defined in the form $E_p(e,f)$: $c^2 = d^3 + ed + f$ over a prime finite field $(d, c) \in W_P^* \times W_P, e, f$ and $4e^3 + 27f^2 \neq 0$ (mod P). Where P is a selected huge prime number, the size of P is ≥ 160 bits. Scalar product is gained by repeated addition e.g. $nP = P + P + P + ... + P(ntimes)$, over a determined t which a point on $E_P(e,f)$ and the multiplier n. The variables (e, f, t, P, n) should be a part of limited field F_P. E is supposed to be the abelian group. Whereas O, is stated as the ID's infinity point.

Definition 2 (Logarithmic issues in ECDLP) ECDLP: is given two specified points over $R, V \in E_P(e,f)$, calculate n a scalar so that $R = nV$. The chances that attacker $\mathcal{A}$ can compute n in polynomial time(T) are described as $Advs_X^{ECDLP}(T) = prob[(X(R, V) = x : xx \in W_P)]$. ECDLP assumption concludes that $Advs_x^{ECDLP}(T) \leq \in$.

Adversarial model

The familiar adversarial model is deliberated in this paper, as specified in [2, 30]. Where the following considerations are followed as per the expertise of the adversary *Advs*:

1. *Advs* have full control over the public communication channel. *Advs* is adept to eliminate, amend, rerun, interrupt or can send a new replicated message.
2. The information stored in the smart card can be excerpted by *Advs*, by doing power analysis.
3. *Advs* can be a deceitful or intruder user or service provider of the system.
4. The identities of registered servers and users are not private but familiar to insiders.
5. The attack on server cannot be launched by *Advs* because the server is assumed to be secured.

Proposed scheme

We propose an anonymous multi-server authentication protocol in this section. Although, proposed protocol brings more computation at server side, but server is usually assumed to have sufficient resources. Therefore, server can easily manages these extra computations in order to lower the computation cost on user side. The scheme is based on multi-server architecture which involves user(U_u), server(S_j) and registration center(RC). RC provides facility for user registration and further helps to give services from server. RC selects its master secret key x to register all users. Like former schemes, the proposed scheme has also three stages: the authentication, registration and password change stage. The proposed protocol is shown in Fig. 2 and described in the below subsections.

Server registration phase

To become legitimate server S_j, the server needs to register with RC by following these steps.

SR Step1: S_j selects his identity ID_j and sends to RC through secure channel.

SR Step2: After receiving ID_j, RC calculates $s = h(ID_j \| x)$, $pk_{S_j} = sP$ and $pk_{RC} = xP$ where x is secret key maintained by RC.

SR Step3: After that, RC sends s, pk_{S_j}, pk_{RC} to server S_j and aborts the registration.

User registration phase

U_u performs the following operations with RC to become the legal user of the network.

UR Step1: User selects his identity ID_u, password PW_u, biometric impression B_u and generates an arbitrary nonce a. Then user determines $M = H(ID_u \| B_u)$,

Server (S_j)	*Registration Centre* ($\mathcal{RC}$)
Each server chooses ID_j	
$\xrightarrow{\{ID_j\}}$	
	Computes $s = h(ID_j \| x)$
	$pk_{S_j} = sP$
	$pk_{RC} = xP$
$\xleftarrow{\{s, pk_{S_j}, pk_{RC}\}}$	

User (U_u)	*Registration Centre* ($\mathcal{RC}$)
Chooses ID_u, PW_u	
Engenders a random nonce a	
Inscribe personal biometric impression B_u	
Determines $M = H(ID_u \| B_u)$	
$TW = h(a \oplus H(B_u \| PW_u))$	
$\xrightarrow{\{ID_u, M, TW\}}$	
	$X_u = h(ID_u \| pk_{RC})$
	$Y_u = X_u \oplus h(M \| TW)$
	$F_u = h(h(ID_u \| TW))$
	Embeds $\{h(), Y_u, F_u\}$ *in* SC_u
$\xleftarrow{\{Smart\ Card\ SC_u\}}$	
U_u takes smart card and embeds a into it	
Now smart card has $\{h(), a, Y_u, F_u\}$	

User (U_u)	*Server* (S_j)
Authentication Phase	
Input its smart-card in specific card-reader	
Enter ID_u and PW_u and biometric impression B_u	
Then SC_u computes	
$TW = h(a \oplus H(B_u \| PW_u))$	
Determine $F_u \stackrel{?}{=} h(h(ID_u \| TW))$	
$M = H(ID_u \| B_u)$	
Generates random number C_u and computes	
$O_p = C_u pk_{S_j} = C_u sP$	
$PID_u = C_u P \oplus ID_u$	
$X'_u = Y_u \oplus h(M \| TW)$	
$DID_u = h(ID_u \| X_u \| C_u P)$	
$\xrightarrow{M_1 = \{PID_u, DID_u, O_p\}}$	
	$s^{-1} O_p = C_u P$
	$ID_u = C_u P \oplus PID_u$
	$X_u = h(ID_u \| pk_{RC})$
	$DID_u \stackrel{?}{=} h(ID_u \| X_u \| C_u P)$
	Generates random number D_j
	$T_u = h(ID_u \| X_u)$
	$V_j = D_j \oplus X_u$
	$Q_{uj} = h(ID_u \| T_u \| C_u P \| D_j \| X_u \| ID_j)$
$\xleftarrow{M_2 = \{Q_{uj}, V_j\}}$	
$D_j = V_j \oplus X_u$	
$h(ID_u \| h(ID_u \| X'_u) \| C_u P \| D_j \| X'_u \| ID_j) \stackrel{?}{=} Q_{uj}$	
$SK_{uj} = h(ID_u \| C_u P \| D_j \| X'_u \| ID_j)$	
$Z_{uj} = h(SK_{uj} \| ID_u \| D_j \| X'_u \| ID_j)$	
$\xrightarrow{M_3 = \{Z_{uj}\}}$	
	Checks $SK_{uj} = h(ID_u \| C_u P \| D_j \| X_u \| ID_j)$
	$h(SK_{uj} \| ID_u \| C_u P \| X_u \| ID_j) \stackrel{?}{=} Z_{uj}$

$\longleftarrow$ *Common Exchanged Key* $= SK_{uj} = h(ID_u \| C_u P \| D_j \| X_u \| ID_j)$ $\longrightarrow$

Fig. 2 Proposed Scheme

$TW = h(a \oplus H(B_u \| PW_u))$ and sends ID_u, M, TW to RC for completing the registration.

UR Step2: After that RC determines $X_u = h(ID_u \| pk_{RC})$, $Y_u = X_u \oplus h(M \| TW)$ and $F_u = h(h(ID_u \| TW))$, then RC stores $h(), Y_u, F_u$ in smart card and sends (SC_u) towards U_u.

UR Step3: U_u further adds a number a into SC_u. Now smart card have $\{h(), Y_u, F_u, a\}$.

Login and authentication phase

In this phase, authenticated access is granted to user U_u for accessing service providers S_j. U_u and S_j authenticate themselves in following steps.

LAP Step1: U_u inputs ID_u, password PW_u and scan biometric impression in scanner. Then smart card determines $TW = h(a \oplus H(B_u \| PW_u))$ and checks whether $F_u \stackrel{?}{=} h(h(ID_u \| TW))$. If yes, then determines $M = H(ID_u \| B_u)$, U_u creates a random number C_u and computes $O_p = C_u pk_{S_j} = C_u sP$, $PID_u = C_u P \oplus ID_u$, $X_u^{'} = Y_u \oplus h(M \| TW)$ and $DID_u = h(ID_u \| X_u^{'} \| C_u P)$. Then U_u sends $M_1 = PID_u, DID_u, O_p$ to S_j.

LAP Step2: After receiving $M_1 = PID_u, DID_u, O_p$, S_j using his secret key s computes $s^{-1} O_p = C_u P$, $ID_u = C_u P \oplus PID_u$ and $X_u = h(ID_u \| pk_{RC})$. After that S_j checks $DID_u \stackrel{?}{=} h(ID_u \| X_u \| C_u P)$. If it holds true, then RC creates arbitrary nonce D_j and determines $T_u = h(ID_u \| X_u)$, $V_j = D_j \oplus X_u$ and $Q_{uj} = h(ID_u \| T_u \| C_u P \| D_j \| X_u \| ID_j)$. Subsequently, S_j sends a message $M_2 = Q_{uj}, V_j$ to U_u.

LAP Step3: U_u determines $D_j = V_j \oplus X_u$ after receiving M_2 and checks $h(ID_u \| h(ID_u \| X_u^{'}) \| C_u P \| D_j \| X_u^{'} \| ID_j) \stackrel{?}{=} Q_{uj}$.

LAP Step4: If the $h(ID_u \| h(ID_u \| X_u^{'}) \| C_u P \| D_j \| X_u^{'} \| ID_j) \stackrel{?}{=} Q_{uj}$ holds true, U_u further determines $SK_{uj} = h(ID_u \| C_u P \| D_j \| X_u^{'} \| ID_j)$ and computes $Z_{uj} = h(SK_{uj} \| ID_u \| D_j \| X_u^{'} \| ID_j)$. U_u sends $M_3 = Z_{uj}$ towards S_j so that it can check the challenge based on D_j.

LAP Step5: After getting M_3, the server S_j determines $SK_{uj} = h(ID_u \| C_u P \| D_j \| X_u \| ID_j)$. After that, it justifies the equation i.e. $h(SK_{uj} \| ID_u \| C_u P \| X_u \| ID_j) \stackrel{?}{=} Z_{uj}$. Finally, on successful justification, server exchanges the session-key SK with user as $h(ID_u \| C_u P \| D_j \| X_u \| ID_j)$. The description of this protocol can be endorsed from Fig. 2.

Password changing phase

U_u may change his password into another new password (PW_u^n) by using these steps. These steps are as follows:

PC Step1: Initially, user input identity ID_u^*, password(PW_u^*) and scan biometric impression factor after inserting smart card (SC) into reader. After that, SC determines $TW = h(a \oplus h(B_u \| PW_u))$ and justify $F_u \stackrel{?}{=} h(h(ID_u \| TW))$. If it holds true then user will follow next step.

PC Step2: Subsequently, SC determines $TW = h(a \oplus h(B_u \| PW_u))$ and calculates $X_u = h(ID_u \| TW), Y_u* = X_u \oplus h(M \| TW)$.

PC Step3: Afterwards, when user will change password (PWi^n). The smart card then determines $TW = h(a \oplus h(B_u \| PW_u^n))$, $Y_u^n = X_u \oplus h(M \| TW')$, and $F_u^n = h(h(ID_u \| TW))$.

PC Step4: Then the values X_u, Y_u, and F_u are changed by X_u^n, Y_u^n, F_u^n in smart card.

Revocation/re-registration phase

In this section, we show that if U_u's smart card has been stolen or his account has been revoked then he can request for reregistration. For this purpose he must follow subsequent steps:

RP Step1: (U_u) engenders a random number a^*, a new password PW_u^*, and biometric B_u^* of his/her own choice. Then calculates $M^* = H(ID_u \| B_u^*)$ and $TW^* = h(a^* \oplus H(B_u^* \| PW_u^*))$ and submits request message $\{ID_u, M^*, TW^*\}$ to the registration centre ($\mathcal{RC}$) through a secure path.

RP Step2: On receiving request message $\{ID_u, M^*, TW^*\}$ from (U_u), $\mathcal{RC}$ will first verify whether (U_u) is already a registered user or not from the verifier table. If a match is not found in the database, the $\mathcal{RC}$ will reject the request.

RP Step3: $\mathcal{RC}$ then embeds the security parameters $\{h(), Y_u^*, F_u^*\}$ *in a new* SC_u^* into the smart card and sends the new smart card to the user (U_u) through secure path.

RP Step4: U_u takes new smart card SC_u^* and embeds a^* into it. The phase is shown in Fig. 3.

$User\ (U_u)$	$Registration\ Centre\ (\mathcal{RC})$
Chooses ID_u, PW_u^* Engenders a random nonce a^* Inscribe new personal biometric impression B_u^* Determines $M^* = H(ID_u \| B_u^*)$ $TW^* = h(a^* \oplus H(B_u^* \| PW_u^*))$	
$\xrightarrow{\{ID_u, M^*, TW^*\}}$	
	$X_u = h(ID_u \| pk_{RC})$ $Y_u^* = X_u \oplus h(M^* \| TW^*)$ $F_u^* = h(ID_u \| TW^*)$ Embeds $\{h(), Y_u^*, F_u^*\}$ *in a new* SC_u^*
$\xleftarrow{\{Smart\ Card\ SC_u^*\}}$	
U_u takes new smart card SC_u^* and embeds a^* into it *Now smart card has* $\{h(), a^*, Y_u^*, F_u^*\}$	

Fig. 3 Revocation phase

Security analysis

In this section, informal and formal security analysis are presented. The security analysis highlights that the proposed scheme is safe and secure against various possible attacks.

Informal security

In this section, a comprehensive informal security analysis of contributed protocol is presented.

Correct notion of user anonymity

In several authentication schemes for multi-server environment, the server is usually unable to identify the identity of a user requesting for login. In our view, such notion of perfect anonymity is erroneous and not desirable in any environment, because if the server is unable to know a user's identity, he will be unable to provide the specific services to the user. In fact in this, any user can continue to get the services provided by the service provider even if he is not registered to the network or his lease has been expired. However, in proposed protocol, instead of user's identity ID_u, a dynamic-pseudo identity PID_u is sent during each authentication request message, to S_j. Furthermore, user's identity ID_u can only be extracted using server's private key s. In addition, by analyzing two different session, an adversary will remain unable to guess whether the same user has initiated session. Hence, in this way our introduced protocol provides user's anonymity and untraceability.

Replay attack

In this flaw, the retrieved messages are restored without endure transformation to deceive any legitimate user [31–34]. Adversary can get the parameters $PID_u, DID_u, O_p, Q_{uj}, T_u, V_j$ and try to endure these parameters in request to forge the legal member. However, if an adversary retrieves contents, he cannot initiate an attack because C_u and D_j is created by legitimate member for every session. Similarly, if an adversary endeavors to replay M_1= PID_u, DID_u, O_p toward server, server verifies the validity of user in M_3, in reply to the challenge based on D_j. Synchronously, the legitimate user validates S_j in M_2 to response to the M_1 based challenge C_u. Hence the contributed protocol thwart replay attack.

Stolen smart card attack with offline dictionary

In stolen smart card attack with offline dictionary, the attacker tries different sequences of dictionary ingredients using stolen SC credentials [35–37]. An attacker may attempts to exploit with its feasible parameters of SC i.e $h()$, Y_u, F_u. For estimating the PW_u from Y_u and F_u parameters, adversary needs to perceive ID_u, a and B_u to estimate PW_u from TW where $TW = h(a \oplus h(B_u \| PW_u))$. Furthermore, this attack cannot initiate in polynomial time using smart card.

Known-key security
Known-key security provides the confidentiality of private keys even with exposed session key for a particular session [38, 39]. Given that the specific session-key $SK_{uj} = h(ID_u \| C_u P \| D_j \| X_u \| ID_j)$ does not hold $U_u's$ password $PW_u's$ as a parameter. Owing it to, the adversary may not discover the parameters from derived session key. Hence, the contributed protocol offers known-key security.

Mutual authentication
Mutual authentication is provided by the enhanced scheme because the legitimate participants verify each other and thus it ensure mutual authentication strongly. This property makes our protocol secure and provides the early detection of possible attacks like replay attacks.

Masquerading attack
According to this attack, an attacker can masquerade one member of a specific session, if it reveals another member's key of the current session. The contributed protocol is immune to key-compromise impersonation threat in contrary to scheme, [23] as the contents of stolen card will not help the attacker to get other constructive parameters, such as X_u. Hence, the attacker cannot obtain newly generated Q_{uj} factor and ultimately impersonation attack cannot be initiated.

Stolen verifier attack
The adversary misuses valued data which is stored at server's side and user's privates like passwords or other parameter, masquerade as legal users. The contributed protocol offers mutual authentication without maintaining repository on S_j and RC's side. This shows that our scheme is withstand stolen verifier attack.

Password guessing attack
The guessing attack is applicable, if an adversary accesses the parameters $PID_u, DID_u, O_p, Q_{uj}, T_u, V_j$ on little analysis of any open channel. Nonetheless, an adversary cannot extract the password, after all it is not use as a factor for the computation of any contents, hence it minimizes the chances of estimating the consistent factors.

Modification attacks
The adversary changes the retrieving parameters and submit to promise party. In case, the scheme is designed to resist against modification threat. If the adversary attempts to change the public contents $PID_u, DID_u, O_p, Q_{uj}, T_u, V_j$, adversary will not able reassemble following parameters PID_u, DID_u, O_p by introducing recent session arbitrary variables, since to assemble these parameters acquires the information of secret key and X_u which knows to legal member. Consequently, the legal member can expose any venomous member easily. So, the enhanced scheme can easily discourage this attack.

Formal security analysis

We have described model of security for presented protocol in this section. Furthermore, using given model of security the presented protocol is proved safe against known attacks. At the end, the proposed protocol is described to fulfill all the necessary requirements that relates to the security of the presented protocol.

Theorem THM1 *Consider D_i as a uniformly distributed dictionary consists of various possible passwords. $|D|$ denotes the size of D_i. Consider A as an adversary against semantic security within a time bound t. Suppose a ECCDH problem stands, then we have*

D_i is considered as evenly distributed dictionary which consists of numerous passwords that can be possible. The size of D_i is denoted by $|D|$. A is considered as an adversary against syntactic security in a time bound t. If a *ECCDH* problem occurs, then we have

$$\begin{aligned} A_{\Pi,D}(A) \leq & \frac{(q_{hsh}+q_{exe})^2}{2p} + \frac{q_{hsh}^2}{p} \\ & + \frac{q_{hsh}}{p} + q_{hsh}A_{\Pi}^{ECCDH}(A) \\ & + \frac{q_{hsh}}{p} + \frac{q_{snd}^2}{D}. \end{aligned} \tag{1}$$

where the possibility of solving the *ECCDH* problem by A, is denoted by A_{Π}^{ECCDH}. The number of Execute, Random-oracle and Send query are $\{q_{exe}, q_{hsh}, q_{snd}\}$, respectively.

Proof In order to give the proof of Theorem THM1, six composite games are considered from game G_1 to G_6. The game will be started where the real attack is simulated and a game will be ended where adversary A has no advantage. The possibility of successfully guessing the random bit b in test-query by A is denoted by Suc_i for each game G_i, where $1{\leq}i{\leq}6$.

GAME G_1: In this random oracle model, the real attacks are simulated with the help of this game. In game G_1, every instance like U_u, S_j and RC will be modeled as authentic executions. As per the definition of $Suc1$, we get following equation.

$$A_{\Pi,D}^{ECCDH}(A) = 2Pr(Suc1) - 1. \tag{2}$$

GAME G_2: Multiple oracles like hash oracle h Execute, Corrupt, Reveal, Send and Test are simulated with G_2. Hash oracle is simulated by game G_2 by maintaining a hash list h_{list}, h_{list} comprises on queries entries as (input, output). When a hash query is answered by hash oracle, then it returns the corresponding output if there is any existing query (input, output) in h_{list}, else it will return value from 0, 1. Moreover, corrupt, reveal, send and Test queries will be run as real attacks. Thread model is used to specify the actual actions of all these queries. This simulation indicates that game G_2 is perfectly secured from the real attacks. Thus, we have

$$Pr(Suc2) = Pr(Suc1) \tag{3}$$

GAME G_3: This game consists on all possible executions of ROM as elaborated in game G_2 except that it will be discarded when some collision occured in the simulation of all hash queries, that are inquired by the adversary A. So, this game helps to avoid from collision to be occurred in ciphertext, password and output of Send-queries. By the definition of birthday paradox, the chances of occuring collision in hash oracle is $\frac{q_{exe}^2}{2p}$. Thats why the chances of occuring collision in game G_3 is $\frac{(q_{hsh}+q_{exe})^2}{2p}$. For this simulations, we achieved following equation

$$|Pr(Suc3) - Pr(Suc2)| \leq \frac{(q_{hsh} + q_{exe})^2}{2p} + \frac{q_{exe}}{2p}. \tag{4}$$

GAME G_4: This game consists on all possible executions of ROM as elaborated in game G_3 but it will be discarded after the successful guessing of X_u by adversary A without asking the hash oracle h. This game is similar all previous games unless the instances Π_U^i and Π_S^j S_j reject the actual authentication value. From game G_4, we get following equation

$$|Pr(Suc4) - Pr(Suc3)| \leq \frac{q_{hsh}}{p} \tag{5}$$

GAME G_5: This game indicates that if adversary guesses the session key directly without knowing and inquiring about hash oracle h then this game will be terminated. It enables the session key to be independent with $\{PW_u, B_u\}$ and random numbers as well as point multiplication C_u, D_j, P. G_4. This game will be aborted after the inquiring common value X_u. Thus, $A_\Pi^{ECCDH}(A) \leq \frac{1}{q_{hsh}} |Pr(Suc5) - Pr(Suc4)| - \frac{1}{p}$ and we have

$$|Pr(Suc5) - Pr(Suc4)| \leq q_h A_\Pi^{ECCDH}(A) + \frac{q_{hsh}}{p}. \tag{6}$$

GAME G_6: This game consists on all possible executions of ROM as elaborated in game G_5 except the rule if follow in Test query. G_5 will be aborted when A queries about hash oracle that is identical values C_u, D_j, P. The chance of adversary A getting the correct session-key by hash-query is at most $\frac{q_{hsh}^2}{2p}$. Thus, we have

$$|Pr(Suc6) - Pr(Suc5)| \leq \frac{q_{hsh}^2}{2p}. \tag{7}$$

Until adversary A does not enter correct value into the random oracle h, the random oracle will remain indistinguishable against real attack. That's why A does not have any advantage of identifying the legal session key from random oracle attempt. Furthermore, when corrupt query is performed, not more than 3 queries can be performed simultaneously. It means that if smart card corrupt and biometric corrupt (Π_U^i,3), (Π_U^i,4) are performed then password corrupt ((Π_U^i,2)) cannot be performed this is the reason that success rate of off-line password guessing attack is $\frac{q_{snd}^2}{D}$. By combining all the equations from G_1 to G_6, we get following equation

$$A_{\Pi,D}(A) \leq \frac{(q_{hsh}+q_{exe})^2}{2p} + \frac{q_{hsh}^2}{p} + \frac{q_{hsh}}{p} + q_h A_{\Pi}^{ECCDH}(A) + \frac{q_{hsh}}{p} + \frac{q_{snd}^2}{D}. \quad (8)$$

Performance analysis

In this section, the robustness of proposed protocol is assessed with respect to other schemes [20–23, 25] based on multi server architecture. The security traits and the scrutiny of defending to numerous attacks for different schemes are described in Table 2, in which the proposed protocol is signified as a strong corroborated key-agreement in contrast to former schemes. Table 2 presents the analysis of our schemes with related schemes [20–23, 25]. As per the analysis, we can conclude that our protocol is more secure than [20–23, 25]. All these protocols depend upon hash-based symmetric cryptography and similar in nature.

Table 2 Comparison of security parameters

Scheme:	Proposed	Liao and Wang [20]	Hsiang and Shih [21]	Lee et al. [22]	Chen and Lee [23]	Irshad et al. [25]
Immune to smart card stolen attack	Yes	Yes	Yes	Yes	No	Yes
Efficient password modification	Yes	Yes	Yes	No	No	Yes
Ensuring anonymity	Yes	Yes	Yes	Yes	Yes	No
Immune to insider attack	Yes	No	Yes	Yes	Yes	Yes
Immune to trace attack	Yes	Yes	Yes	Yes	No	Yes
Immune to impersonation attack	Yes	No	No	No	No	Yes
Support mutual authentication	Yes	No	No	No	Yes	Yes
Repair ability	Yes	Yes	No	No	Yes	Yes
Supports session key security	Yes	Yes	Yes	Yes	No	Yes
Immune to offline password guessing attack	Yes	Yes	Yes	No	Yes	Yes
Immune to KCI attack	Yes	Yes	No	Yes	No	Yes

Table 3 Comparison of energy and computational, communication and storage costs

Protocols	Computational cost	Energy(mJ)	Communication cost($bits$)	Storage cost($bits$)
Proposed	$8T_h + 7T_h + 1T_H = 0.0120ms$	0.91	1920	1440
Liao and Wang [20]	$9T_h + 9T_h = 0.0117ms$	0.98	2944	1024
Hsiang and Shih [21]	$9T_h + 12T_h = 0.0132ms$	1.10	3296	1440
Lee et al. [22]	$9T_h + 10T_h = 0.0125ms$	1.05	3296	1440
Chen and Lee [23]	$12T_h + 10T_h = .0146ms$	1.24	3252	1440
Irshad et al. [25]	$13T_h + 13T_h + 3T_H = 0.0169ms$	1.64	3872	1952

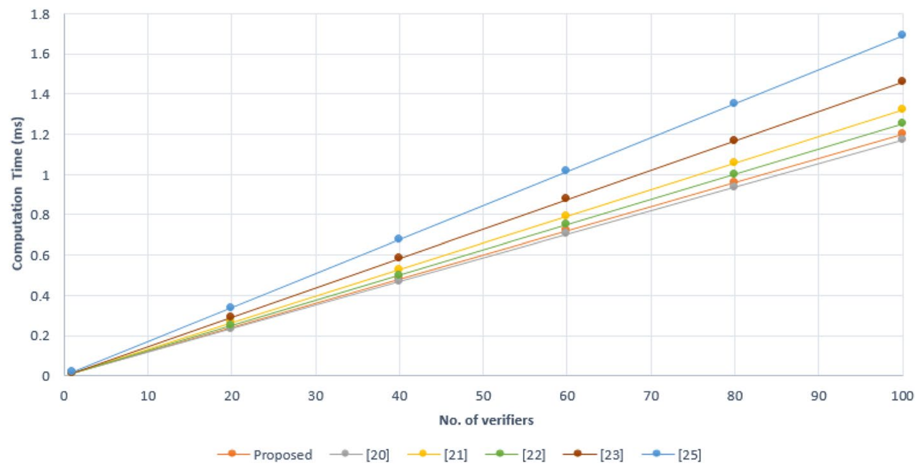

Fig. 4 Computation cost comparison

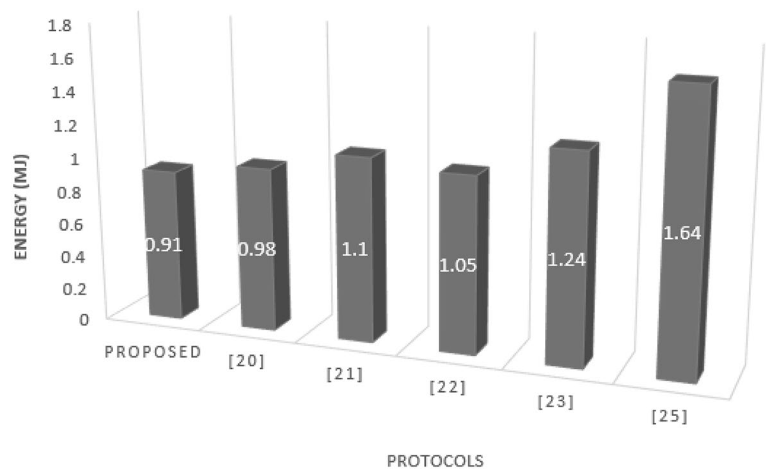

Fig. 5 Analysis of energy of proposed and other protocols

Later, the performance analysis of our authenticated protocol in terms of cost has been analyzed. The specification and description for the implementation is as follows; the implementation of the cryptographic functions ($T_{\oplus}, T_{\|}, T_{h(.)}, P_m$) is done by using py-crypto library inside ubuntu 19.04, with 16.0 GB RAM and 3.60 GHz processor core $i7$ with the help of python programming language. The execution of authentication scheme is done under same assumptions for 10 times by averaging. Some functions like ($T_{\|}, T_{\oplus}$) have not been considered because they acquires negligible execution time. The execution time for $h(.)$, $H(.)$ and point multiplication operations is 0.0120 ms, 0.015 ms and 0.02957 ms, respectively. The communication, energy requirements, storage and computation cost of our scheme with respect to related protocols is presented in Table 3. The time for execution of considered cryptographic functions are assumed as follows:

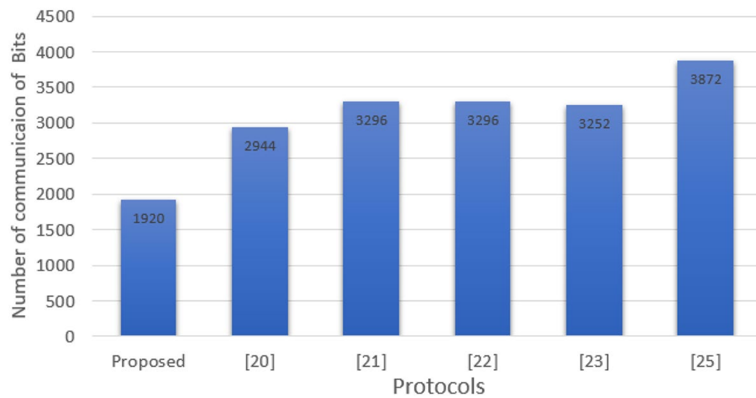

Fig. 6 Analysis of communication cost between proposed and related protocols

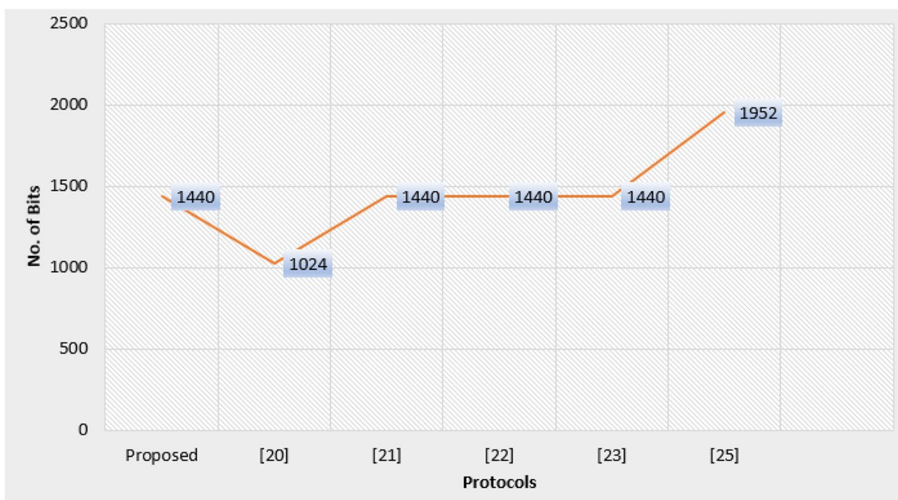

Fig. 7 Analysis of storage cost of proposed and other protocols

- Execution time for one way hash function is $E_h = 0.0120ms$.
- Execution time for one way bio-hash function is $E_H = 0.015ms$.
- Execution time for point multiplication $E_{pm} = 0.02957ms$.

It is observed the computation cost of our proposed scheme is higher than [20–23, 25] schemes but it offers aided security features. Furthermore, the mandatory security objectives are achieved by our protocol in less cost than Hsiang and Shih's scheme. Moreover, the proposed protocol (contrasting with former protocols) is secure to smart card stolen, password guessing and insider attacks.

We have determined cost comparison in Table 3, which are later elaborated by drawing Figs. 4, 5, 6 and 7. The cost of computation for proposed and relevant schemes is showcased in Fig. 4. The number of verifiers of our proposed and existing protocols are shown horizontally and required computation time according to the number of verifiers is shown vertically in the graph. It can be observed that computation cost of our protocol is far less than the related schemes..

Energy consumption can be calculated as $E_c = T_{cc}P_{CPU}$, where T_{cc} is the total computation cost for a single hash function (0.054 *mJ*), P_{CPU} is the maximum power (65 *W*) of CPU and E_c is the energy consumption [40]. Power consumption can be used to give a rough estimate of energy consumed during computation. Moreover, we have examined the protocol with respect to energy consumption by considering computation cost of energy for SHA-1 as 0.54 *mJ* for single byte [41] shown in Fig. 5. By Considering this, the consumption of energy for the [20–23, 25] and our scheme amounts to 1.64 *mJ*, 0.98*mJ*, 1.10 *mJ*, 1.05 *mJ*, 1.24 *mJ* and 0.91*mJ*, respectively. The final energy consumption determined values of proposed and related schemes are given in Table 3. Hence, it can be calculated that the energy consumption of proposed scheme is less than related schemes.

The assumptions made for determining the communication and storage cost are as follows: 160 bits are reserved for random nonce, timestamps, password and identity, 256 bits are for one way hash function and for public key, 512 bits. The calculations of storage and communication cost of our and related schemes on the basis of above mentioned assumptions are presented in Table 3.

The cost of communication for proposed and relevant schemes is presented in Fig. 6. The proposed and related schemes are given horizontally, while the required number of communication bits are shown vertically in the graph. It is observed that the number of communication bits of proposed scheme is slightly greater than related schemes but our scheme provides more security traits. The storage cost of proposed and related schemes is displayed in Fig. 7. The vertically labeled values on the graph are for the required number of storage bits, while proposed and related schemes are listed horizontally.

The storage bits of our scheme is slightly greater than the related protocol. This is just because of providing more security features for making secure protocol. After analyzing Tables 2 and 3, we can say that the computation time of our scheme is less than the related schemes and also provides more security traits with slightly higher communication and storage costs.

Conclusion

The robustness of multi-server authentication is observed as an important requisite for the current remote based authentication paradigm. Recently, extensive research has been conducted for developing robust authentication protocols for multi-server authentication environment. In this paper, we proposed an anonymous multi-server authentication scheme. The flaws of previous schemes are kept in mind in order to develop the proposed scheme with enhanced security features. The analysis of performance evaluation and formal security is also described in this paper against various schemes. This analysis also shows that our scheme provides more security features.

Acknowledgements
Not applicable

Authors' contributions
MAA, ZG and KM analyzed the requirement of the security for Multi-server Infrastructure, designed the framework, conducted the experiment and drafted the manuscript. Also, KM gave full support in conducting the experiment and assisted in draft work and revised the manuscript. SK contributed by reviewing the work done and in revising the content of the manuscript. In addition, all the work is done in SK's supervision. KA has helped us in major revision according to the comments of reviewers. CMC coordinated the whole study. All authors read and approved the final manuscript.

Availability of data and materials
It is theoretical work no data is needed.

Competing interests
The authors proclaim, they have no competing interests.

Author details
[1] Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus, Sahiwal 57000, Pakistan. [2] Department of Mathematics, Chaudhary Charan Singh University, Meerut, Uttar Pradesh 250004, India. [3] Department of Computer Science and Engineering, Bhagwati Institute of Technology and Science, Ghaziabad 201302, India. [4] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China.

Received: 3 January 2020 Accepted: 17 April 2020
Published online: 15 May 2020

References

1. Lin I-C, Hwang M-S, Li L-H (2003) A new remote user authentication scheme for multi-server architecture. Fut Gen Comput Syst 19(1):13–22
2. Juang W-S (2004) Efficient password authenticated key agreement using smart cards. Comput Secur 23(2):167–173
3. Chen C-M, Xiang B, Liu Y, Wang K-H (2019) A secure authentication protocol for internet of vehicles. IEEE Access 7:12047–12057
4. Kumari S, Chaudhary P, Chen C-M, Khan MK (2019) Questioning key compromise attack on ostad sharif et al authentication and session key generation scheme for healthcare applications. IEEE Access 7:39717–39720
5. Kumari S, Li X, Wu F, Das AK, Choo K-KR, Shen J (2017) Design of a provably secure biometrics-based multi-cloud-server authentication scheme. Fut Gen Comput Syst 68:320–330
6. Kumari S, Khan MK, Atiquzzaman M (2015) User authentication schemes for wireless sensor networks: A review. Ad Hoc Netw 27:159–194
7. Kumari S, Li X, Wu F, Das AK, Arshad H, Khan MK (2016) A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps. Fut Gen Comput Syst 63:56–75
8. Mahmood K, Naqvi H, Alzahrani BA, Mehmood Z, Irshad A, Chaudhry SA (2018) An ameliorated two-factor anonymous key exchange authentication protocol for mobile client-server environment. Int J Commun Syst 31(18):3814
9. Chen C-M, Huang Y, Wang K-H, Kumari S, Wu M-E (2020) A secure authenticated and key exchange scheme for fog computing. Enterprise Information Systems pp 1–16
10. Chen C-M, Wang K-H, Yeh K-H, Xiang B, Wu T-Y (2019) Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications. J Ambient Intell Human Comput 10(8):3133–3142
11. Kumari S, Obaidat M, Wei F et al (2020) Gateway-oriented two-server password authenticated key exchange protocol for unmanned aerial vehicles in mobile edge computing. IET Communications
12. Mahmood K, Arshad J, Chaudhry SA, Kumari S (2019) An enhanced anonymous identity-based key agreement protocol for smart grid advanced metering infrastructure. Int J Commun Syst 32(16):4137
13. Chaudhry SA, Shon T, Al-Turjman F, Alsharif MH (2020) Correcting design flaws: An improved and cloud assisted key agreement scheme in cyber physical systems. Computer Communications
14. Mahmood K, Rehman A, Chaudhary P, Li X, Wu F (2020) Revised anonymous authentication protocol for adaptive client-server infrastructure. Int J Commun Syst 2020:4253
15. Lee W-B, Chang C-C (2000) User identification and key distribution maintaining anonymity for distributed computer networks. Comput Syst Sci Eng 15(4):211–214
16. Wu T-S, Hsu C-L (2004) Efficient user identification scheme with key distribution preserving anonymity for distributed computer networks. Comput Secur 23(2):120–125
17. Tsaur W-J (2001) A flexible user authentication scheme for multi-server internet services. In: International Conference on Networking, pp. 174–183 . New York, Springer
18. Li L-H, Lin L-C, Hwang M-S (2001) A remote password authentication scheme for multiserver architecture using neural networks. IEEE Trans Neural Netw 12(6):1498–1504
19. Chang C-C, Lee J-S (2004) An efficient and secure multi-server password authentication scheme using smart cards. In: 2004 International Conference on Cyberworlds, pp. 417–422 . IEEE
20. Liao Y-P, Wang S-S (2009) A secure dynamic id based remote user authentication scheme for multi-server environment. Comput Stand Interf 31(1):24–29
21. Hsiang H-C, Shih W-K (2009) Improvement of the secure dynamic id based remote user authentication scheme for multi-server environment. Comput Stand Interf 31(6):1118–1123
22. Lee C-C, Lin T-H, Chang R-X (2011) A secure dynamic id based remote user authentication scheme for multi-server environment using smart cards. Exp Syst Appl 38(11):13863–13870
23. Chen C-T, Lee C-C (2015) A two-factor authentication scheme with anonymity for multi-server environments. Secur Commun Netw 8(8):1608–1625
24. Chang C-C, Lee C-Y (2013) A smart card-based authentication scheme using user identify cryptography. IJ Netw Secur 15(2):139–147
25. Irshad A, Naqvi H, Ashraf Chaudhary S, Usman M, Shafiq M, Mir O, Kanwal A (2018) Cryptanalysis and improvement of a multi-server authenticated key agreement by chen and lee's scheme. Inform Technol Contr 47(3):431–446
26. Kumari S, Das AK, Li X, Wu F, Khan MK, Jiang Q, Islam SH (2018) A provably secure biometrics-based authenticated key agreement scheme for multi-server environments. Multim Tools Appl 77(2):2359–2389

27. Li X, Peng J, Obaidat MS, Wu F, Khan MK, Chen C (2020) A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems. IEEE Syst J 14(1):39–50
28. Li X, Wu F, Kumari S, Xu L, Sangaiah AK, Choo K-KR (2019) A provably secure and anonymous message authentication scheme for smart grids. J Parallel Distrib Comput 132:242–249
29. Altaf I, Arslan Akram M, Mahmood K, Kumari S, Xiong H, Khurram Khan M (2020) A novel authentication and key-agreement scheme for satellite communication network. Trans Emerg Telecommun Technol. https://doi.org/10.1002/ett.3894
30. Arshad R, Ikram N (2013) Elliptic curve cryptography based mutual authentication scheme for session initiation protocol. Multim Tools Appl 66(2):165–178
31. Amin R, Biswas G (2015) Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment. Wireless Pers Commun 84(1):439–462
32. Amin R, Islam SH, Biswas G, Khan MK, Leng L, Kumar N (2016) Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. Comput Netw 101:42–62
33. Irshad A, Chaudhry SA, Xie Q, Li X, Farash MS, Kumari S, Wu F (2018) An enhanced and provably secure chaotic map-based authenticated key agreement in multi-server architecture. Arab J Sci Eng 43(2):811–828
34. Ravanbakhsh N, Nazari M (2018) An efficient improvement remote user mutual authentication and session key agreement scheme for e-health care systems. Multim Tools Appl 77(1):55–88
35. Islam SH, Obaidat MS, Amin R (2016) An anonymous and provably secure authentication scheme for mobile user. Int J Commun Syst 29(9):1529–1544
36. Tsai J-L (2008) Efficient multi-server authentication scheme based on one-way hash function without verification table. Comput Secur 27(3–4):115–121
37. Shamshad S, Mahmood K, Kumari S (2020) Comments on 'a multi-factor user authentication and key agreement protocol based on bilinear pairing for the internet of things'. Wireless Personal Communications pp 1–4
38. Amin R, Islam SH, Biswas G, Khan MK, Kumar N (2018) A robust and anonymous patient monitoring system using wireless medical sensor networks. Fut Gen Comput Syst 80:483–495
39. Kalra S, Sood SK (2015) Secure authentication scheme for iot and cloud servers. Perv Mob Comput 24:210–223
40. Alshahrani M, Traore I (2019) Secure mutual authentication and automated access control for iot smart home using cumulative keyed-hash chain. J inform Secur Appl 45:156–175
41. Potlapally NR, Ravi S, Raghunathan A, Jha NK (2006) A study of the energy consumption characteristics of cryptographic algorithms and security protocols. IEEE Trans Mob Comput 5(2):128–143

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.