Salem *et al. Journal of Cloud Computing* (2023) 12:128
https://doi.org/10.1186/s13677-023-00499-3

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

AMAKAS: Anonymous Mutual Authentication and Key Agreement Scheme for securing multi-server environments

Fatty M. Salem[1*], Maha Safwat[2], Rasha Fathy[2] and Shahira Habashy[2]

Abstract

The rapid growth of Internet users was the motivation of the emerge appearance of new computing models such as cloud computing, fog computing and edge computing. For this reason, the multi-server's architecture has been introduced to extend scalability and accessibility. To ensure that these servers can only be accessed by the authorized users, many authentication and key agreement schemes have been introduced for multi–server environments. In this paper, we propose an anonymous mutual authentication and key agreement scheme for multi-server architecture based on elliptic curve cryptography to achieve the required security services and resist the well-known security attacks. Furthermore, formal and informal security analysis is conducted to prove the security of the proposed scheme. Moreover, we provide a performance comparison with related work in terms of computational cost, communication cost and the number of messages transferred on the public channel. This performance comparison clearly shows that the proposed scheme is highly efficient in terms of computation, communication cost and security analysis as compared to other related schemes which makes the proposed scheme more suitable and practical for multi-server environments than other related schemes.

Keywords Multi–server environment, Elliptic curve cryptography, Mutual authentication, Anonymity, Un-traceability

Introduction

The multi-server environment was established as a result of the rapid increase in internet users and Internet of Things (IoT). A multi-server environment is a sort of server infrastructure that makes use of multiple physical servers to give consumers access to numerous services and applications. The key benefit of using a multi-server system is that it can provide a higher level of availability, reliability, and security than a single-server environment. Additionally, because the load may be distributed across numerous servers, a multi-server architecture can provide a higher level of performance than a single-server environment. However, secure and efficient communication between the concerned parties has grown more vital in multi-server environment especially in areas including e-commerce and distributed storage systems.

Many security requirements must be achieved in multi-server environments such as mutual authentication between the user and the server, user anonymity, user intractability and forward secrecy. Moreover, there are many types of attacks that must be resisted in multi-server environment such as impersonation attack, replay attack, insider attack, stolen card attack, man-in-the-middle attack, and known session specific temporary information attack. To communicate securely and effectively over an unsecure network, a shared session key must be negotiated and agreed between the involved parties first. The only remedy for such negotiations is to

*Correspondence:
Fatty M. Salem
faty_ahmed@h-eng.helwan.edu.eg
[1] Department of Electronics and Communications Engineering, Helwan University, Helwan, Cairo, Egypt
[2] Department of Computers and Systems Engineering, Helwan University, Helwan, Cairo, Egypt

use authentication-and-key-agreement protocols. The first password authentication using insecure communication is proposed by Lamport [1] as the most simple and practical method for authenticating a user from remote servers. However, Lamport's scheme [1] could not resist insider attack once the password file stored in the server is compromised. To overcome this limitation, many two-factor authentication schemes have been proposed based on smart cards in which important secret parameters are stored [2–4]. The main drawback of the two-factor authentication schemes is the power analysis attack on a stolen smart card which may lead the scheme to be exposed to offline password attack. As a result, research has shifted to three-factor authentication techniques based on biometrics [5–7].

Elliptic Curve Cryptography (ECC) was employed in two-factor and three-factor authentication protocols [8–11] in order to gain the advantages of ECC properties of creating small size keys with high security efficiency [12]. Some multi-sever authentication protocols employed registration center to be involved not only in registration phase but also in authentication phase between user and server in order to decrease the computation load on user to overcome the limitation resources of the user [10, 13, 14]. However, involving the registration center also in authentication phase between user and server adds overload on the registration center which causes delay of registration center response.

Motivation

The rapid increase of internet users and IoT makes the current research concerned in multi-server environment in which authentication and key agreement are the main goals to securely offer several services and applications. Numerous existing schemes are proposed to provide authentication and key agreement in multi-server environment using different methods such as password-based authentication [1], smart cards-based authentication [4, 15, 16], three-factor authentication [6, 7, 13], dynamic ID-based authentication [17, 18], and ECC-based authentication [9, 12]. However, most of the existing schemes can't achieve some security services like mutual authentication and user untraceability and can't resist different types of attacks. On the other side, the scheme that succeeded in providing secure communications is at the expense of high computation cost and communication overhead. Motivated by the existing studies and the need of secure multi-server environment, designing a lightweight authenticated key agreement scheme with a small number of messages of small number of bits is imperatively needed to resist security threats, reduce communication overhead, and to meet the limitations of devices with low computation capabilities.

Contributions

We summarize our significant and key contributions in the field of multi-server environments as follows:

- Firstly, a multi-server environment is considered, and then the Elliptic Curve Cryptography is employed to design the proposed Anonymous Mutual Authentication and Key Agreement Scheme (AMAKAS) for securing multi-server environments.
- The proposed AMAKAS scheme guarantees the security requirements of multi-server environments and withstands against various types of attacks in multi-server environments.
- The proposed AMAKAS scheme enables users to mutually authenticate with servers without involving the registration center in the authentication phase.
- The performance of the proposed AMAKAS scheme is outperformed than the related schemes.

Road map of the paper

The remaining section of the paper is structured as follows: In "Related work" section, related work is reviewed, "System model and threat model" section depicts the system model and the threat model, while "The proposed AMAKAS scheme" section introduces the proposed anonymous mutual authentication and key agreement scheme. "Security analysis" section passes through a security analysis of the proposed scheme. The security and performance comparison with other related schemes is demonstrated in "Security and performance comparisons" section. Finally, the paper is concluded in "Conclusion" section.

Related work

In 2016, Chang et al. proposed a scheme based on smart card and biometrics [15]; this scheme can resist offline password guessing and stolen card attack, but it could not resist user impersonation attack. In 2017, Quan et al. proposed a biometrics-based scheme [16] to overcome the shortcoming of Chang et al.'s scheme [15] to resist the impersonation attack. In the same year, Jangirala et al. proposed a remote user authentication scheme based on dynamic ID using smart cards [17] in which the user is free to choose his login credentials; However, Sahoo et al. [18] proved that Jangirala et al.'s scheme [17] failed to attain mutual authentication as it claimed and failed to resist user impersonation attack, forgery attack, and replay attack. Additionally, Sahoo et al. [18] proposed an improved two-factor dynamic ID based scheme to overcome the shortcoming of Jangirala et al. [17]; however, Sudhakar et al. [19] analyzed Sahoo et al.'s scheme [18] and proved that it still cannot resist

replay and user impersonation attack. Shunmuganathan [20] proposed a lightweight two factor-based scheme to overcome the drawbacks of Sahoo et al. [18], but it failed to achieve user anonymity nor user-un-tractability as a result. Moreover, Shunmuganathan's scheme [20] has high computations at server and registration center.

Kuo-Hui Yeh proposed a novel multi-server-based authentication scheme [21]; however, Truong et al. [22] proved that Kuo-Hui Yeh's scheme [21] failed to achieve mutual authentication and session-key agreement. Hence, Truong et al. [22] proposed an improved ECC based scheme to overcome the shortcoming of Yeh's scheme [21]. However, Yan et al. [23] observed that Truong et al.'s scheme [22] could not resist impersonation attack. Hence, Yan et al. [23] they proposed a scheme to overcome Truong et al.'s scheme [22]. However, Yan et al.'s scheme [23] still cannot achieve user anonymity and needs synchronization nodes to resist replay attack. Additionally, Yan et al.'s scheme [23] can't resist man-in-the-middle attack and known session specific temporary information attack.

In 2020, Akram et al. [24] proposed a three factor ECC-based authentication scheme that can achieve mutual authentication and user anonymity, and could resist replay, impersonation and password guessing attack. However, on the other hand, Akram et al.'s scheme [24] has a very high computational time due to using the ECC multiplicative inverse. In 2021, Amintoosi et al. [25] proposed an ECC-based three factor authentication scheme which is capable to achieve mutual authentication, user anonymity and forward secrecy, but it could not achieve user un-tractability and could not resist sever impersonation attack. Wang et al. [26] proposed a biometric-based multi-server authentication scheme using elliptic curve cryptosystem to achieve authentication; however, Wu et al. [27] demonstrated that Wang et al.'s scheme [26] can't resist user impersonation attacks, server impersonation attacks, and known session-specific temporary information attacks. Both schemes [26, 27] suffered from high computations at registration center side as registration center is involved in authentication phase.

In 2022, Truong et al. proposed a three factor-based authentication scheme [28] in which registration center is a party in authentication phase to decrease the computations at user side where the user's resources are limited compared to registration center. Truong et al.'s scheme [28] could achieve mutual authentication and user anonymity, and also could resist both user and server impersonation attack, replay attack, man-in-the-middle attack, and known session specific temporary information attack. However, Truong et al.'s scheme [28] failed to achieve user un-tractability and suffers from high load computation at registration center side.

Guo et al. proposed biometric-based authentication scheme using public key encryption [29] to achieve authentication; however, Chen et al. [30] demonstrated that Guo et al.'s scheme [29] can't resist user impersonation attack and replay attack. Additionally, Chen et al. [30] proposed a threeFactor authentication scheme to overcome the drawbacks of Guo et al.'s scheme [29]; but Chen et al. [30] failed to resist server impersonation attack; moreover, it needs synchronization nodes to resist replay attack due to using time stamp.

Bae et al. [31] proposed a smart card-based authentication protocol to protect multi-server IoT environment from potential security vulnerabilities; Agarwal et al. [32] demonstrated that Bae et al.'s scheme [31] can't resist user impersonation attack, replay attack, and insider attack. Additionally, Agarwal et al. [32] proposed a threeFactor authentication scheme to overcome the drawbacks of Bae et al.'s scheme [31]; however, Agarwal et al.'s scheme [32] suffers from high computations at server and registration center as well.

Cho et al. [33] proposed an ECC three factor-based authentication scheme to overcome the drawbacks of Sudhakar et al.'s scheme [19], but Cho et al.'s scheme [33] needs synchronization nodes to resist replay attack. Khan et al. [34] proposed an ECC three factor-based authentication scheme for cloud server, but it failed to achieve user un-tractability and it could not resist replay attack.

In 2023, Yao et al. proposed an authentication and key agreement scheme for edge computing in vehicular ad hoc networks (VANETs) [35] based on bilinear map. It could achieve mutual authentication, user anonymity, user un-tractability and forward secrecy. However, Yao et al.'s scheme [35] suffers from high computational time due to employing bilinear map. Also, Also, LAMAS scheme [36] has been proposed for securing fog computing environment; however, the scheme didn't consider the mobility movability of fog users between fog areas.

Ui Haq et al. [37] proposed a hash-based authenticated key agreement scheme using only x-or operations and hash functions. The scheme [37] can achieve user anonymity at a low-cost; however, it can't achieve user un-traceability as the attacker can trace user and link many sessions f the same user by using Ex-OR between the sent parameters of the login request. Moreover, the scheme [37] can't achieve perfect forward secrecy, and it is also vulnerable to replay attacks. Dhillon and Kalra [38] proposed a lightweight three-factor user authentication scheme based on x-or operations and hash functions; however, Lee et al. [39] found that Dhillon and Kalra's scheme [38] can't provide a session key agreement and user un-traceability and can't resist user impersonation attack, replay attack, stolen mobile device attack, and known session-specific temporary information attack.

Additionally, Mahmood et al. [40] proved that Dhillon and Kalra's scheme [38] can't provide user anonymity.

System model and threat model

In this section, the system model and threat model will be demonstrated.

System model

As shown in Fig. 1, multi-servers' architecture consists of three entities which are n users, m servers and the Registration Center (RC).

- In registration phase, RC starts generating the required secret credentials for each user U_i and each S_j as each user and each server must register only once with the registration center. Also, RC stores the U_i's secret parameters generated by RC on a smart card SC and delivers smart card to U_i. Both user registration and server registration are done through a secure channel.
- Once the registration is done, authentication phase started as user authenticate himself by inserting SC into smart card reader and using his login parameters (username, password, and biometric impression) to verify himself. After that, user and server run mutual authentication and key agreement protocol for secure communication between them noting that mutual authentication is done through insecure public channel.
- Once mutual authentication is achieved, any legitimate registered user can connect with any legitimate registered m severs in the network.

Threat model

Assuming that the adversary:

- Has full control over the insecure public communication channel between user and server.
- Can intercept, modify, replay, or even delete messages transmitted through the public channel.
- Can find the secret parameters stored on the smart card using the power analysis attack.
- Can find the password through an offline dictionary attack using parameters which are disclosed from smart card.
- Try to find the current session key and upon revealing the current session key, old session keys can be comprised as well.
- Can run user impersonation attack if user's password or smart card can be accessed.

The proposed AMAKAS scheme

To achieve anonymous mutual authentication and key agreement between user and server in multi-server environments, we proposed a scheme consisting of three phases which are: Registration phase, login phase, and authentication phase.

Registration phase

In this phase, both user and server register with the registration center (RC) as follows:

Server registration

1. Initially, a server S_j registers with the RC by choosing an identity ID_j and sends it to the RC through secure channel.
2. The RC generates a random number e_j and calculates server secret key $ASID_j = h(ID_j||X||e_j)$ where X is the secret key of RC and calculates the server pub-

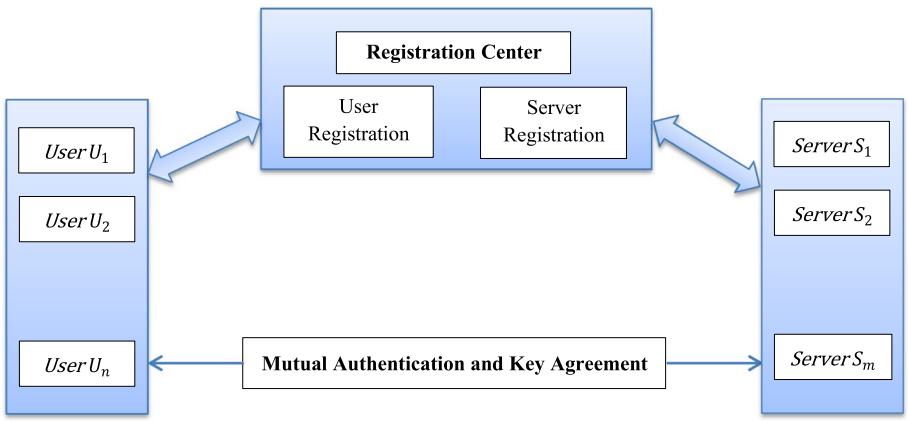

Fig. 1 Multi-server's architecture

lic key $PKS_j = ASID_j.P$ where P is the elliptic curve base point.
3. Finally, the RC sends to each server S_j its own secret key and server public key through a secure channel.

User registration

1. Similarly, each user U_i registers with the RC by selecting the user identity ID_u and password PW_u and describes his biometric impression B_u.
2. User U_i generates random nonce a, calculates $M = H(ID_u||B_u)$ and $TW = h(a \oplus \mathrm{H}(B_u||PW_u))$, and sends $\{ID_u, M, TW\}$ to the RC through secure channel.
3. RC generates random number a_u and calculates $A_{u=}a_u.P$, $X_u = h\left(a_u.PKS_j||ID_u||ASID_j\right)$, $Y_u = X_u \oplus h(M||TW)$, and $F_u = h(h(ID_u||TW))$.
4. Finally, the RC sends $\{A_u, Y_u, F_u\}$ through a secure channel to be printed on the Smart Card (SC).

Login phase

Login and authentication phase is shown in Fig. 2; In login phase, the user U_i logs into a system by taking the subsequent steps:

1. Initially, user U_i inserts the SC into smart card reader and inputs his login parameters$\{ID_u, PW_u, B_u\}$
2. SC calculates $TW = h(a \oplus H(B_u||PW_u))$ and $F_u^* = h(h(ID_u||\mathrm{TW}))$, and compares F_u^* with the stored F_u in the SC.
3. If $F_u^* \neq \mathrm{F}_u$, the session will be discarded; otherwise, SC generates random number C_u and computes $W = C_u.P$, $OP = C_u.PKS_j = C_u.ASID_j.P$, $OPA_u = A_u \oplus OP$, and uses the most significant l-bits of $h(OP)$ to compute $PID_u = ID_u \oplus h(OP)$, $M = H(ID_u||B_u)$, $X_u = Y_u \oplus h(M||TW)$ and $DID_u = h(A_u||X_u||OP)$.
4. Finally, user U_i sends $M_1 = \{W, OPA_u, PID_u, DID_u\}$ to server S_j via public channel.

Authentication phase

In this phase, mutual authentication and key agreement between the user and the server can be achieved by taking the subsequent steps:

1. Upon receiving $M_1 = \{\mathrm{W}, OPA_u, PID_u, DID_u\}$, the server calculates $OP = C_u.PKS_j = W.ASID_j = C_u.P.ASID_j$, $A_u = OPA_u \oplus OP$, and uses the most significant l-bits of $h(OP)$ to compute $ID_u = PID_u \oplus h(OP)$, $X_u = h(A_u.ASID_j||ID_u||ASID_j)$, and $DID_u^* = h(A_u||X_u||OP)$. Then, S_j compares the calculated DID_u^* with the received DID_u.
2. If $DID_u^* \neq DID_u$, the session will be discarded; otherwise, S_j generates random number D_j, and calculates $v_j = D_j \oplus OP$, $SK = h\left(ID_u||OP||D_j||X_u||ID_j\right)$, and $Q_{ju} = h(ID_u||OP||\left|D_j\right|\left|ID_j\right|\left|SK\right)$.
3. The server sends $M_2 = \{Q_{ju}, v_j\}$ to the user U_i via public channel.
4. Upon receiving $M_2 = \{Q_{uj}, v_j\}$, user U_i calculates $D_j = v_j \oplus OP$, $SK = h(ID_u||OP||\left|D_j\right|\left|X_u\right|\left|ID_j\right)$, and $Q_{uj} = h(ID_u||OP||\left|D_j\right|\left|ID_j\right|\left|SK\right)$, and compares the calculated Q_{uj} with the received Q_{ju}.
5. If $Q_{uj} = Q_{ju}$, mutual authentication has been achieved and session key has been agreed between the user U_i and the server S_j; otherwise, the session will be discarded.

Security analysis

This section provides an informal security analysis of the proposed AMAKAS scheme in addition to formal security analysis using Burrows-Abadi-Needham (BAN) logic [41].

Informal security analysis

In this subsection, an informal security analysis will be provided to explain how the proposed AMAKAS scheme achieves the most important security requirements including mutual authentication, user anonymity, un-traceability, and forward secrecy. In addition, we explain how the proposed AMAKAS scheme resists the most known attacks including impersonation attack, replay attack, stolen card attack, man-in-the-middle attack, and known session specific temporary information attack.

Mutual authentication

The proposed AMAKAS scheme achieves mutual authentication since both the legitimate user and the legitimate server can authenticate each other.

The server S_j authenticates the user U_i by computing $DID_u^* = h(A_u||X_u||OP)$ and comparing it with the receivedDID_u inM_1. The user computes $DID_u = h(A_u||X_u||OP)$ by calculating $X_u = Y_u \oplus h(M||TW)$ where Y_u is stored on the SC, calculating $M = H(ID_u||B_u)$ requires knowing the user identity ID_u and biometric impression B_u of the user, and calculating $TW = h(a \oplus \mathrm{H}(B_u||PW_u))$ requires knowing the random number a, the biometric impression B_u, and the user password PW_u which are known only to the legitimate user. Therefore, the server can authenticate the user.

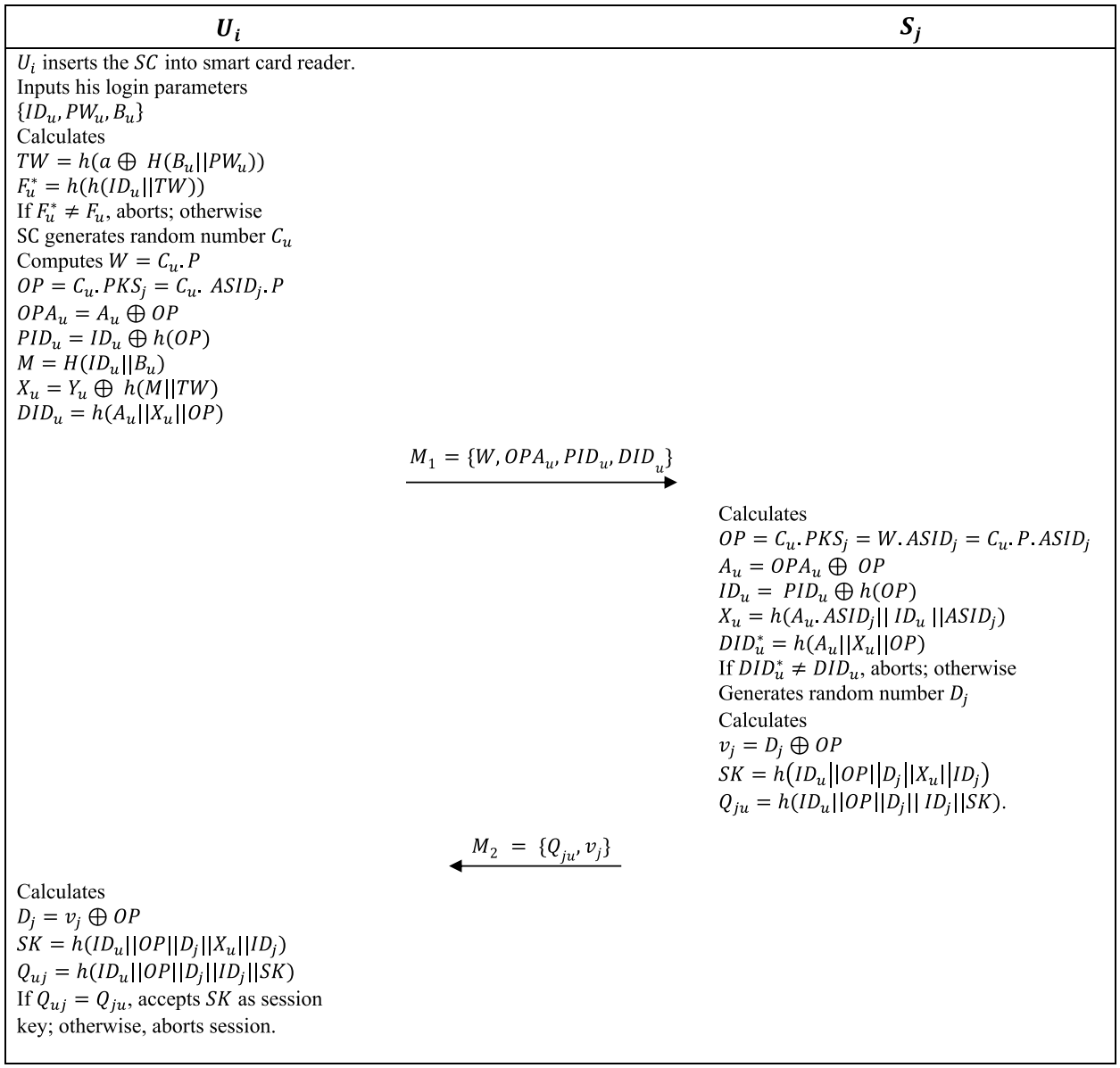

Fig. 2 Login and authentication phase

On the other hand, the user U_i authenticates the server S_j by computing $Q_{uj} = h(ID_u||OP||D_j||ID_j||SK)$ and comparing it with the received Q_{ju} in M_2. The server S_j can obtain $OP = C_u.PKS_j = W.ASID_j$ using the server's private key $ASID_j$ which is known only to the server S_j, and then extract the identity of the user as $ID_u = PID_u \oplus h(OP)$. Thus, the server can authenticate the user.

Therefore, mutual authentication between user and server has been achieved and session key has been agreed on. Furthermore, early detection of any possible replay attack has been ensured.

User anonymity

The proposed AMAKAS scheme can achieve user anonymity as in each authentication message, the user identity ID_u is randomized using $OP = C_u.PKS_j$ where C_u is a random number and hidden through a dynamic-pseudo identity $PID_u = ID_u \oplus h(OP)$. Even if the Adversary A intercepts the transmitted message $M_1 = \{W, OPA_u, PID_u, DID_u\}$, he still cannot extract the user identity ID_u from the dynamic-pseudo identity $PID_u = ID_u \oplus h(OP)$ as the adversary needs first to obtain OP using the server's secret key $ASID_j$ which is unknown to the adversary.

User un-traceability

The proposed AMAKAS scheme can achieve user's untractability as in each login message sent to the server by the user $M_1 = \{\mathrm{W}, OPA_u, PID_u, DID_u\}$,, the user generates a new random number C_u which is used to calculate $W = C_u.P$ and $OP = C_u.PKS_j$, then OP is used to randomize $OPA_u = A_u \oplus OP$, $PID_u = ID_u \oplus h(OP)$, and $DID_u = h(A_u||X_u||OP)$. Hence, the value of the transmitted message $M_1 = \{\mathrm{W}, OPA_u, PID_u, DID_u\}$ is updated in each session. Moreover, if the attacker computes $OPA_u \oplus PID_u$, this will result in $A_u \oplus OP \oplus ID_u \oplus h(OP)$ which is not a fixed value; this is why we used $h(OP)$ to randomize ID_u instead of using OP directly. Thus, even if the Adversary A intercepts the transmitted message $M_1 = \{\mathrm{W}, OPA_u, PID_u, DID_u\}$, he still cannot relate any repeated messages. Therefore, user un-tractability is guaranteed.

Forward secrecy

Forward Secrecy can be achieved in the encryption scheme when producing temporary secret session key uniquely generated for every individual session between user and server. If one of these session keys is compromised, transmitted messages in past sessions will be protected from attacks.

In the proposed AMAKAS scheme, the session keys are independent on each other as in in each session, the session key $SK = h(ID_u||OP||D_j||X_u||ID_j)$ is generated based on new random values of C_u and D_j where D_j is a random number generated by the legitimate server and C_u is a random number generated by the legitimate user as well to compute $OP = C_u.PKS_j$. Therefore, even if the current session key is comprised, the adversary still cannot obtain the previous session keys.

Additionally, assuming that the attacker can get the server's secret key $ASID_j$ and can intercept all transmitted messages $M_1 = \{\mathrm{W}, OPA_u, PID_u, DID_u\}$ and $M_2 = \{Q_{ju}, v_j\}$. Even under these assumptions, without knowing the random number D_j and the value of OP, the attacker will not be able to compromise the messages of previous sessions. Furthermore, the computation to obtain the server's secret key $ASID_j$ is a very complex task due to ECDHP problem.

Impersonation attack

Impersonation attack has two types: user impersonation and server impersonation attack. The proposed AMAKAS scheme can resist both types of impersonation attack.

For the user impersonation attack:

If the adversary aims to impersonate the legitimate user, he has to be capable of generating a valid login message $M_1 = \{\mathrm{W}, OPA_u, PID_u, DID_u\}$. The adversary can generate a random number C_u and calculate W, PID_u, and OPA_u, but he cannot generate $DID_u = h(A_u||X_u||OP)$ as the calculation of $X_u = Y_u \oplus h(M||TW)$ requires knowing $\{Y_u, M, TW\}$, Y_u is a stored value on the smart card, calculating $M = H(ID_u||B_u)$ requires knowing the user identity ID_u and biometric impression B_u of the user, and calculating $TW = h(a \oplus \mathrm{H}(B_u||PW_u))$ requires knowing the random number a, the user password PW_u, and the biometric impression B_u which are known only by the legitimate user. Moreover, password is protected by double hash one way function. Hence, the adversary cannot generate a valid login message M_1, and therefore, the proposed scheme can resist user impersonation attack.

For the server impersonation attack:

The server secret key $ASID_j = h(ID_j||X||e_j)$ is calculated through one way hash function for server ID, secret key of registration center, and the random number e_j generated by the registration center; therefore, $ASID_j$ is only known by the legitimate server. If the adversary aims to impersonate the legitimate server, he has to be capable of generating $M_2 = \{Q_{ju}, v_j\}$, but calculating $v_j = D_j \oplus OP$ requires obtaining the correct value of $OP = C_u.PKS_j = W.ASID_j$ which is based on server's secret key which is known by only legitimate server. Hence, the adversary cannot generate a valid v_j. Similarly for calculating $Q_{ju} = h(ID_u||OP||D_j||ID_j||SK)$, it requires calculating the correct value for OP and the session key $SK = h(ID_u||OP||D_j||X_u||ID_j)$ which is based on calculating $X_u = h(a_i.PKS_j||ID_u||ASID_j)$ which requires knowing the random number a_i generated by the registration center, user ID, and the server's secret key $ASID_j$. Therefore, still only the legitimate server can generate Q_{ju}. Hence, the proposed AMAKAS scheme can resist server impersonation attack.

Replay attack

The proposed AMAKAS scheme can resist replay attack as with each login message $M_1 = \{\mathrm{W}, OPA_u, PID_u, DID_u\}$, generated by the user, a fresh random number C_u is generated. Even if the Adversary could replay M_1, mutual authentication between user and server cannot be achieved as the Adversary does not know the random number C_u; therefore, he cannot compute $OP = C_u.PKS_j$

nor $D_j = v_j \oplus OP$. Hence, he cannot extract the session key $SK = h(ID_u||OP||D_j||X_u||ID_j)$.

Stolen card attack

The proposed AMAKAS scheme can resist the stolen card attack as even if the adversary can steal the SC and extract the stored data on the SC $\{A_u, Y_u, F_u\}$, he still cannot guess the user password nor the user ID since the extracted data are not used in computing the password, and user ID is not included in the extracted data. Therefore, the Adversary cannot generate the login message. Therefore, the proposed AMAKAS scheme can resist stolen card attack.

Man-in-the-middle attack

Between the user and server, a man-in-the-middle attacker pretends to be a node in the middle, but the attacker can't know the password PW_u of the user U_i and can't get his biometric impression B_u, also the attacker can't obtain the secret key $ASID_j$ of server S_j. When the attacker attempts to impersonate each party in this situation, he is unable to generate a valid DID_u as it is computed using $X_u = Y_u \oplus h(M||TW)$ which is locally computed at user U_i using user's password and biometric impression as $TW = h(a \oplus H(B_u||PW_u))$. Additionally, the attacker can't know the shared session key $SK = h(ID_u||OP||D_j||X_u||ID_j)$ as it requires knowing OP and the random number D_j which can't be obtained without knowing the secret key $ASID_j$ of server S_j. Hence, the proposed AMAKAS scheme can resist man-in-the-middle attack.

Known session specific temporary information attack

In this attack, when temporary secret values, such as random numbers, are revealed, an attacker tries to obtain the current session key. After completing the login and authentication phase, if OP and the random number D_j can be obtained, the attacker can compute A_u and ID_u, but it can't compute the session key $SK = h(ID_u||OP||D_j||X_u||ID_j)$ as it depends on X_u which is computed using user's password and biometric impression at user side and using the secret key $ASID_j$ of server S_j at server side. Hence, the proposed AMAKAS scheme can resist known session specific temporary information attack.

Formal security analysis using BAN logic

In this subsection, BAN Logic is used to formally prove the security of the proposed AMAKAS scheme.

Idealization

The idealized messages between the user and the server are listed as follows.

$$M_1 : (U_i \rightarrow S_j) : W, < ID_u >_{OP}, OPA_u, (A_u, OP)_{U_i \overset{X_u}{\leftrightarrow} S_j}$$
$$M_2 : (S_j \rightarrow U_i) : V_i, (ID_u, OP, D_j, ID_j, U_i \overset{SK}{\leftrightarrow} S_j)_{U_i \overset{X_u}{\leftrightarrow} S_j}$$

Assumptions

The assumptions of the proposed scheme to proceed the BAN logic analysis are listed as follows:

$$A_1 : U_i| \equiv \#(C_u)$$
$$A_2 : S_j| \equiv \#(D_j)$$
$$A_3 : U_i| \equiv (U_i \overset{OP}{\leftrightarrow} S_j)$$
$$A_4 : S_j| \equiv (U_i \overset{OP}{\leftrightarrow} S_j)$$
$$A_5 : U_i| \equiv (U_i \overset{X_u}{\leftrightarrow} S_j)$$
$$A_6 : S_j| \equiv (U_i \overset{X_u}{\leftrightarrow} S_j)$$
$$A_7 : S_j| \equiv (U_i \Longrightarrow (OP))$$
$$A_8 : U_i| \equiv (S_j \Longrightarrow (U_i \overset{SK}{\leftrightarrow} S_j))$$
$$A_9 : S_j| \equiv (U_i \Longrightarrow (U_i \overset{SK}{\leftrightarrow} S_j))$$

Goals

The goals that our proposed scheme should be achieved are listed as follows.

$$\text{Goal 1: } U_i| \equiv \left(S_j \overset{X_u}{\leftrightarrow} U_i\right), \#(S_j \overset{X_u}{\leftrightarrow} U_i)$$
$$\text{Goal 2: } U_i| \equiv S_j| \equiv \#(C_u)$$
$$\text{Goal 3: } S_j| \equiv U_i| \equiv \#(D_j)$$

Analysis

The following steps are taken to perform the BAN logic proof of our suggested scheme.

Step 1: From message M_2, we obtain:

$$U_i \triangleright V_i, (ID_u, OP, D_j, ID_j, U_i \overset{SK}{\leftrightarrow} S_j)_{U_i \overset{X_u}{\leftrightarrow} S_j}$$

Step 2: From the assumption A_5, we obtain:

$$U_i| \equiv (U_i \overset{X_u}{\leftrightarrow} S_j)$$

Step 3: From M_2 and A_5, and applying the message-meaning rule, we obtain:

$$\frac{U_i| \equiv \left(U_i \overset{X_u}{\leftrightarrow} S_j\right), U_i| \lhd V_i, (ID_u, OP, D_j, ID_j, U_i \overset{SK}{\leftrightarrow} S_j)_{U_i \overset{X_u}{\leftrightarrow} S_j}}{U_i| \equiv S_j| \sim (V_i, ID_u, OP, D_j, ID_j, (U_i \overset{SK}{\leftrightarrow} S_j))}$$

Step 4: From A_1, A_2, step 2, and applying nonce verification rule, we obtain,

$$\frac{U_i| \equiv \#(C_u), S_j| \sim (V_i, ID_u, OP, D_j, ID_j, (U_i \overset{SK}{\leftrightarrow} S_j))}{U_i| \equiv S_j| \equiv \left(U_i \overset{SK}{\leftrightarrow} S_j\right)}$$

Step 5: from A_8, step 4, and applying the jurisdiction rule, we obtain:

$$\frac{U_i| \equiv \left(S_j| \Longrightarrow \left(U_i \overset{SK}{\leftrightarrow} S_j\right)\right), U_i| \equiv S_j \equiv \left(U_i \overset{SK}{\leftrightarrow} S_j\right)}{U_i| \equiv (U_i \overset{SK}{\leftrightarrow} S_j)}$$

Step 6: From A_1, A_2, step 4, and applying the freshness conjuncatenation rule, we obtain:

$$\frac{U_i| \equiv \#(C_u)}{U_i| \equiv \#\left(C_u, \left(U_i \overset{SK}{\leftrightarrow} S_j\right)\right)}$$

Step 7: From step 5 and step 6, we obtain:

$$U_i| \equiv \left(U_i \overset{SK}{\leftrightarrow} S_j\right) and U_i| \equiv \#\left(U_i \overset{SK}{\leftrightarrow} S_j\right)$$

Hence, Goal 1 has been achieved.

Step 8: From step 2, A_2, and applying the nonce verification rule, we obtain:

$$\frac{U_i| \equiv \#(C_u), S_j \sim \left(V_i, ID_u, OP, D_j, ID_j, \left(U_i \overset{SK}{\leftrightarrow} S_j\right)\right)}{U_i| \equiv S_j| \equiv (D_j)}$$

Step 9: From step 8, A_8, and applying the jurisdiction rule, we obtain:

$$\frac{U_i| \equiv \left(S_j| \Longrightarrow \left(U_i \overset{SK}{\leftrightarrow} S_j\right)\right), U_i| \equiv S_j| \equiv (D_j)}{U_i| \equiv (D_j)}$$

Step 10: From step 9,A_2, step 4 and applying the freshness conjuncatenation rule, we obtain:

$$\frac{U_i| \equiv \#(C_u)}{U_i| \equiv \#(C_u, (D_j))}$$

Thus, $U_i| \equiv \#(D_j)$ and Goal 3 has been achieved.

Step 11: From A_6, and applying the message-meaning rule, we obtain:

$$\frac{U_i| \equiv \left(U_i \overset{X_u}{\leftrightarrow} S_j\right), S_j| \triangleleft W, < ID_u >_{OP}, OPA_u, (A_u, OP)_{U_i \overset{X_u}{\leftrightarrow} S_j}}{U_i| \sim (W, < ID_u >_{OP}, OPA_u, A_u, OP)}$$

Step 12: From A_1, step 11, and applying the nonce verification rule, we obtain:

$$\frac{U_i| \equiv \#(C_u), U_i \sim (W, < ID_u >_{OP}, OPA_u, A_u, OP)}{S_j| \equiv U_i| \equiv (OP)}$$

Step 13: From A_7, step 12, and applying the jurisdiction rule, we obtain:

$$\frac{S_j| \equiv (U_i \Longrightarrow (OP)), U_i| \equiv S_j| \equiv (OP)}{S_j| \equiv (OP)}$$

Noting that $OP = C_u.PKS_j$

Step 14: From A_1, step 11, step 13, and applying the freshness conjuncatenation rule, we obtain:

$$\frac{U_i| \equiv \#(C_u)}{S_j| \equiv \#(C_u)}$$

Therefore, $S_j| \equiv \#(C_u)$ and Goal 2 has been achieved.

Security and performance comparisons

In this section, the security and performance of the proposed AMAKAS scheme are compared with the existing related schemes. The performance will be evaluated in terms of computation cost and communication overheads.

Security comparison

Table 1 provides a summarized analysis for the security features of the proposed AMAKAS scheme while comparing it with some related schemes [23–25, 28, 38]. From Table 3, we can observe that the schemes in [23, 38] cannot achieve user anonymity, and the schemes [23, 25, 28, 38] are unable to achieve user un-traceability. Moreover, the schemes [23, 25, 38] cannot resist man-in-the-middle attack, and none of the schemes in [23–25, 38] can resist server impersonation attack or known session specific temporary information attack. It can be seen that the lightweight authentication scheme in [38] can't resist against several attacks including user impersonation

Table 1 Security comparison

Computation Party	[23]	[24]	[25]	[28]	[38]	The proposed AMAKAS scheme
Mutual authentication	√	√	√	√	√	√
User anonymity	x	√	√	√	x	√
User Un-traceability	x	√	x	x	x	√
Forward secrecy	√	√	√	√	√	√
User impersonation attack	√	√	√	√	x	√
Sever impersonation attack	x	x	×	√	√	√
Replay attack	√	√	√	√	x	√
Stolen card attack	√	√	√	√	x	√
Man-in-the-middle attack	x	√	x	√	x	√
Known session specific temporary information attack	x	x	x	√	x	√

attack, replay attack, stolen card attack, man-in-the-middle attack, and known session specific temporary information attack. However, it is obvious that the proposed AMAKAS scheme can achieve mutual authentication, user anonymity, user un-traceability and forward secrecy. In addition, our scheme can resist user and server impersonation attacks, replay attack, stolen card attack, man-in-the-middle attack, and known session specific temporary information attack.

Computation cost comparison

In this section, we present an analysis for the computational cost of the proposed AMAKAS scheme compared with the related schemes [23–25, 28] that can provide equipollent security requirements.

Table 2 shows the execution time of the required cryptographic operations for the comparison between the proposed AMAKAS scheme, and the other related schemes as computed in [42] using a machine with E2200 2.20 GHz Intel Pentium CPU, 2 GB of RAM, and a 32-bit Ubuntu 12.04.1 LTS operating system. During calculating the computational cost, we are considering the following operations: T_h is execution time of one-way hash function, T_p is execution time of ECC scalar multiplication, T_{inv} is the execution time multiplicative inverse over ECC, T_m is the execution time of point addition, T_{SED} is the execution time of symmetric key encryption/decryption, T_{AED} is the execution time of ECC encryption/decryption, and T_F is the execution time of fuzzy extraction. The pre-mentioned operations are calculated while using a machine with E2200 2.20 GHz Intel Pentium CPU, 2 GB of RAM, and a 32-bit Ubuntu 12.04.1 LTS operating system.

Table 3 shows the computation cost of login and authentication phases for the proposal schemes compared to schemes [23–25, 28]. We can observe that scheme [24] consumes the highest execution time during login and authentication phase, it costs 190.189E + 06 ms due to the complex operation of computing the multiplicative inverse over ECC. Scheme [23] consumes time of executing 10 hash functions, 4 ECC scalar multiplication, and 5-point addition operations which totally costs 9.071 ms. Scheme [25] consumes time of executing 14 hash function, 3 ECC scalar multiplication, and one ECC encryption/decryption operations which totally costs

Table 2 The execution time of the required cryptographic operations

Notation	Description	Execution time (ms)
T_h	Time of one-way hash function.	0.0023
T_m	Time of point addition.	0.0288
T_P	Time of ECC Scalar multiplication.	2.226
T_{inv}	Time of multiplicative inverse over ECC.	190.189E + 06
T_{SED}	Time of symmetric key encryption/decryption.	0.0046
T_{AED}	Time of ECC encryption/decryption.	3.85
T_F	Time for fuzzy extraction	2.226

Table 3 Computation cost comparison

Computation Party	[23]	[24]	[25]	[28]	The proposed AMAKAS scheme
User	$4T_h+2T_P+2T_m$	$11T_h+2T_P$	$10T_h+2T_P+1T_{AED}$	$8T_h+2T_P+3T_m+1T_F$	$10T_h+2T_P$
(ms)	4.5188	4.4773	8.325	6.7828	4.475
Server	$6T_h+2T_P+3T_m$	$6T_h+1T_{inv}$	$4T_h+1T_P$	$5T_h+3T_P+4T_m$	$5T_h+2T_P$
(ms)	4.5522	≈190.189E+06	2.2352	6.8116	4.4635
Total	$10T_h+4T_P+5T_m$	$17T_h+2T_P+1T_{inv}$	$14T_h+3T_P+1T_{AED}$	$13T_h+5T_P+7T_m+1T_F$	$15T_h+4T_P$
(ms)	9.071	190.189E+06	10.5602	13.5944	8.9385

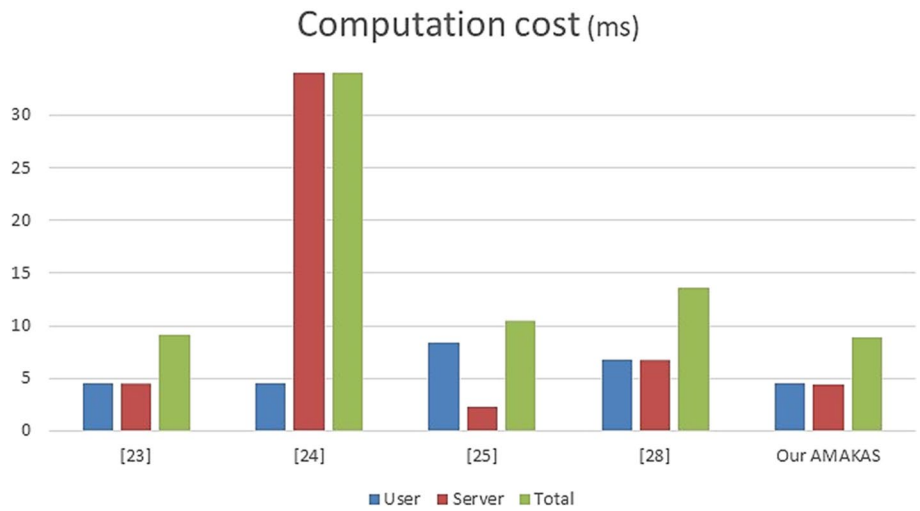

Fig. 3 The comparison of computation cost

10.5602 ms. Scheme [28] consumes execution time of 13 one-way hash functions, 5 ECC Scalar multiplications, 7-point addition operations and one fuzzy extraction operations which totally costs 13.5944 ms. Finally, it is obvious that the lowest computation cost can be offered by the proposed scheme as the proposed scheme consumes the time of executing 15 one-way hash functions and 4 ECC Scalar multiplication operations which total costs 8.9385 ms. The comparison of computation cost is also graphically shown in Fig. 3.

Hence, the proposed scheme is highly efficient in terms of computation cost as compared to other related schemes which makes the proposed AMAKAS scheme more suitable and practical for multi-server environments than other related schemes.

Communication overhead comparison

The number of communication messages is shown in Table 4. It is obvious that the proposed AMAKAS scheme and the scheme in [24] require only 2 messages to complete login and authentication phase, however, the schemes in [23, 25, 28] require 3 messages to complete the same phases.

Table 4 Number of communication messages

Scheme	[23]	[24]	[25]	[28]	The proposed AMAKAS scheme
No of Messages	3	3	2	3	2

In Table 5, we compared the communication overhead of the proposed scheme and that of the schemes [23–25, 28], where the bit size of random number, user's identity, timestamp, ECC point, and hash output (using SHA-1 as h(·)) are 160, 160, 32, 320, 160 bits, respectively. We can observe that the proposed AMAKAS scheme requires 1280 bits to transmit M_1 and M_1, which is the less than the schemes in [23, 25, 28], while it is slightly higher than the scheme in [24] which is a little cost compared to the advantages of the proposed scheme in terms of the computation cost over the scheme in [24] which requires 190.189E+06 ms to execute login and authentication phase.

Table 5 Communication overhead comparison

Scheme	[23]	[24]	[25]	[28]	The proposed AMAKAS scheme
No of bits	1440	1120	1344	1760	1280

As a result, we can state that our proposed scheme is more appropriate for multi-server environments in terms of performance and security.

Conclusion

In this paper, we have proposed a lightweight ECC based mutual authentication and key agreement scheme in multi-server environments. The proposed AMAKAS scheme employed ECC in order to obtain the advantage of ECC properties of creating small size keys with high security efficiency. The security analysis shows that the proposed AMAKAS scheme can achieve mutual authentication, user anonymity and untractability, and forward secrecy. In addition, the proposed AMAKAS scheme can resist replay attack without the need for synchronization nodes, user and server impersonation attack, stolen card attack, man-in-the-middle attack, and known session specific temporary information attack. Moreover, the proposed AMAKAS scheme decreases the computational and communication cost the other related schemes with only two messages of exchange to provide anonymous authentication and key agreement. These advantages make the proposed AMAKAS scheme more suitable and practical for multi-server environments than other related schemes.

Acknowledgements
The authors acknowledge and appreciate the anonymous reviewers for their valuable comments which improved this paper.

Authors' contributions
Authors' contributions Conceptualization: Fatty M. Salem. Investigation: Fatty M. Salem, Rasha Fathy. Methodology: Fatty M. Salem, Maha Safwat. Verification: Fatty M. Salem, Maha Safwat. Supervision: Shahira M. Habashy.Writing – original draft: Fatty M. Salem, Maha Safwat. Writing – review & editing: Fatty M. Salem, Shahira M. Habashy.

Funding
Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB). Publication fees is funded by Technology & Innovation Funding Authority (STDF) in cooperation with Egyptian Knowledge Bank (EKB).

Availability of data and materials
Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Consent has been granted by all authors.

Competing interests
The authors declare no competing interests.

Received: 15 April 2023 Accepted: 5 August 2023
Published online: 30 August 2023

References

1. Lamport L (1981) Password authentication with insecure communication. Commun ACM 24:770–772
2. Wang P, Zhang Z, Wang D (2018) Revisiting anonymous two-factor authentication schemes for multi-server environment. Information and communications security ICICS 2018. Lecture notes in computer science, vol 11149. Springer, Cham, pp 805–816
3. Xu G, Qiu S, Ahmad H, Xu G, Guo Y, Zhang M, Xu H (2018) A multi-server two-factor authentication scheme with un-traceability using elliptic curve cryptography. Sensors 18(7):2394
4. Ul haq I, Wang J, Zhu Y (2020) Secure two-factor lightweight authentication protocol using self-certified public key cryptography for multi-server 5G networks. J Netw Comput Appl 161:102660
5. Sahoo S, Mohanty S, Majhi B (2017) A lightweight three factor-based authentication scheme for multi-server environment using smart cards. 7th international conference on communication and network security, Tokyo Japan, 24–26 November 2017. pp 43–47
6. Sudhakar T, Natarajan V (2019) A new three-factor authentication and key agreement protocol for multi-server environment. Wirel Netw 26:4909–4920
7. Patela C, Joshib D, Nishant D, Veeramuthuc A, Jhaveri R (2020) An enhanced approach for three factor remote user authentication in multi - server environment. J Intell fuzzy Syst 39:8609–8620
8. Zhang L, Tang S, Chen J, Zhu S (2015) Two-factor remote authentication protocol with user anonymity based on elliptic curve cryptography. Wireless Pers Commun 81:53–75
9. Lee CC, Li CT, Weng CY, Jheng JJ, Zhang XQ, Zhu YR (2013) Cryptanalysis and improvement of an ECC-based password authentication scheme using smart cards. 5th international symposium, CSS 2013, Zhangjiajie, China, 13–15 November 2013, Lecture notes in computer science (LNCS, volume 8300). pp 338–348
10. Wang CH, Hsu KC (2019) Enhancing biometric and mutual verification in multi-server three-factor user remote authentication scheme with elliptic curve cryptography. 7th international conference on communications and broadband networking, April 2019. pp 46–5
11. Kumar R, Gupta MK, Kumari S (2021) ECC-based three-factor authentication scheme for multi-server environment. In ISIC. pp 158–163
12. Roy S, Khatwani C (2017) Cryptanalysis and improvement of ECC based authentication and key exchanging protocols. Cryptography 1(1):9
13. Ali R, Pal AK (2018) An efficient three factor-based authentication scheme in multiserver environment using ECC. Int J Commun Syst 31:e3484. https://doi.org/10.1002/dac.3484
14. Wang F, Xu G, Wang C, Peng J (2019) A provably secure biometrics-based authentication scheme for multiserver environment. Secur communication networks 4:1–15
15. Chang CC, Hsueh WY, Cheng TF (2016) An advanced anonymous and biometrics-based multi-server authentication scheme using smart cards. Int J Netw Secur 18(4):1010–1021
16. Quan C, Lee H, Kang D, Kim J, Cho S, Won D (2018) Cryptanalysis and improvement of an advanced anonymous and biometrics-based multi-server authentication scheme using smart cards. International conference on human factors in Cybersecurity, 17 – 21 July 2017, The Westin Bonaventure Hotel, Los Angeles, California, USA, 593. pp 62–71
17. Jangirala S, Mukhopadhyay S, Das AK (2017) A multi-server environment with secure and efficient remote user authentication scheme based on dynamic ID using smart cards. Wireless Pers Commun 95(3):2735–2767
18. Sahoo SS, Mohanty S, Majhi B (2018) An improved and secure two-factor dynamic ID based authenticated key agreement scheme for multi-server environment. Wireless Pers Commun 101:1307–1333

19. Sudhakar T, Natarajan V, Gopinath M, Saranyadevi J (2020) An enhanced authentication protocol for multiserver environment using password and smart card. Wireless Pers Commun 115:2779–2803
20. Shunmuganathan S (2021) A reliable lightweight two factor mutual authenticated session key agreement protocol for multiserver environment. Wireless Pers Commun 121:2789–2822
21. Yeh KH (2014) A provably secure multi-server based authentication scheme. Wireless Pers Commun 79(3):1621–1634
22. Truong TT, Tran MT, Duong AD, Echizen I (2017) Provable identity-based user authentication scheme on ECC in multi-server environnement. Wireless Pers Commun 95:2785–2801
23. Zhao Y, Li S, Jiang L (2018) Secure and efficient user authentication scheme based on password and smart card for multiserver environment. Secur Commun Netw. https://doi.org/10.1155/2018/9178941
24. Akram MA, Ghafar Z, Mahmood K, Kumari S, Agarwal K, Chen CM (2020) An anonymous authenticated keyagreement scheme for multiserver infrastructure. Hum - centric Comput Inform Sci 10:22
25. Amintoosi H, Nikooghadam M, Kumari S, Kumar S, Chen CM (2021) TAMA: three-factor authentication for multi-server architecture. Hum - centric Comput Inform Sci 11:39
26. Wang F, Xu G, Wang C, Peng J (2019) A provably secure biometrics-based authentication scheme for multiserver environment. Secur Commun Netw. https://doi.org/10.1155/2019/2838615
27. Wu T, Yang L, Lee Z, Chen CM, Pan JS, Hafizul Islam SK (2021) Improved ECC-based three-factor multiserver authentication scheme. Secur Commun Netw 2021:1. https://doi.org/10.1155/2021/6627956
28. Truong TT, Tran MT, Duong AD, NguyenPham PN, Nguyen HA, Nguyen TN (2022) Provable user authentication scheme on ECC in multiserver environment. J Supercomput 79:725–761
29. Guo H, Wang P, Zhang X, Huang Y, Ma F (2017) A robust anonymous biometric-based authenticated key agreement scheme for multi-server environments. PLoS ONE 12(11):e0187403. https://doi.org/10.1371/journal.pone.0187403
30. Chen R, Mou Y, Zhang M (2022) A novel Threefactor authentication scheme with high security for multiserver environments. Wireless Pers Commun 124:763–781
31. Bae WI, Kwak J (2017) Smart card-based secure authentication protocol in multi-server IoT environment. Multimedia Tools Appl 79:15793–15811
32. Agarwal K, Gupta AK, Kumari S, Sain M (2022) A secure authentication scheme for teleservices using multi-server architecture. Electronics 11(18):2839
33. Cho Y, Oh J, Kwon D, Son S, Yu S, Park Y (2022) A secure three-factor authentication protocol for E-governance system based on multiserver environments. IEEE Access 10:74351–74365
34. Khan N, Zhang J, Jan SU (2022) A robust and privacy-preserving anonymous user authentication scheme for public cloud server. Secur Commun Netw. https://doi.org/10.1155/2022/1943426
35. Yao M, Gan Q, Wang X, Yang Y (2023) A key-insulated secure multi-server authenticated key agreement protocol for edge computing-based VANETs. Internet Things 21:100679
36. Hamada M, Salem S, Salem F (2022) LAMAS: lightweight anonymous mutual authentication scheme for securing fog computing environments. Ain Shams Eng J 13(6):101752. https://doi.org/10.1016/j.asej.2022.101752
37. Ui Haq I, Wang J, Zhu Y, Maqbool S et al (2021) An efficient hash-based authenticated key agreement scheme for multi-server architecture resilient to key compromise impersonation. Digit Commun Netw 7(1):140–150
38. Dhillon PK, Kalra S (2017) Secure multi-factor remote user authentication scheme for internet of things environments. Int J Commun Syst 30(16):e3323
39. Lee H, Kang D, Ryu J, Won D, Kim H, Lee Y (2020) A three-factor anonymous user authentication scheme for internet of things environments. J Inform Secur Appl 52:102494. https://doi.org/10.1016/j.jisa.2020.102494
40. Mahmood K, Akram W, Shafiq A, Altaf I, Lodhi M, Islam SK (2020) An enhanced and provably secure multi-factor authentication scheme for internet-of-multimedia-things environments. Comput Electr Eng 88:106888. https://doi.org/10.1016/j.compeleceng.2020.106888
41. Burrows M, Abadi M, Needham RM (1989) A logic of authentication. Proc R Soc Lond A 426(1871):233–271
42. Abbasinezhad-Mood D, Ostad-Sharif A, Nikooghadam M (2020) Novel anonymous key establishment protocol for isolated smart meters. IEEE Trans Industr Electron 67(4):2844–2851

Publisher's Note