PLOS ONE

RESEARCH ARTICLE

An efficient dynamic ID-based remote user authentication scheme using self-certified public keys for multi-server environments

Shudong Li[1,2], **Xiaobo Wu**[3], **Dawei Zhao**[4]*, **Aiping Li**[2], **Zhihong Tian**[1]*, **Xiaodong Yang**[5]

1 Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China, **2** College of Computer, National University of Defense Technology, Hunan Changsha, China, **3** School of Software Engineering, Yantai Vocational College, Shandong Yantai, China, **4** Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China, **5** College of Computer Science and Engineering, Northwest Normal University, Gansu Lanzhou, China

* zhaodw@sdas.org (DZ); tianzhihong@gzhu.edu.cn (ZT)

Citation: Li S, Wu X, Zhao D, Li A, Tian Z, Yang X (2018) An efficient dynamic ID-based remote user authentication scheme using self-certified public keys for multi-server environments. PLoS ONE 13(10): e0202657. https://doi.org/10.1371/journal.pone.0202657

Editor: Hua Wang, Victoria University, AUSTRALIA

Received: December 18, 2017

Accepted: July 15, 2018

Published: October 9, 2018

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This work is supported by the National Natural Science Foundation of China (61672020, 61662069, 61472433, 61702309, and 61572153, to S. L., X.Y., A.L., D.Z., and Z.T), the Project funded by China Postdoctoral Science Foundation (2013M542560, 2015T81129 to S.L.), National Key Research and Development Plan (2017YFB0801804, 2017YFB0802204 to A.L.), and A Project of Shandong Province Higher Educational

Abstract

Recently, Li et al. proposed a novel smart card and dynamic ID-based remote user authentication scheme for multi-server environments. They claimed that their scheme can resist several types of attacks. However, through careful analysis, we find that Li et al.'s scheme is vulnerable to stolen smart card and off-line dictionary attacks, replay attacks, impersonation attacks and server spoofing attacks. By analyzing other similar schemes, we find that a certain type of dynamic ID-based multi-server authentication scheme in which only hash functions are used and whereby no registration center participates in the authentication and session key agreement phase faces difficulties in providing perfectly efficient and secure authentication. To compensate for these shortcomings, we propose a novel dynamic ID-based remote user authentication scheme for multi-server environments based on pairing and self-certified public keys. Security and performance analyses show that the proposed scheme is secure against various attacks and has many excellent features.

Introduction

With the rapid development of network technologies, increasingly more people are beginning to use networks to acquire various services such as on-line financial information, on-line medical information, on-line shopping, on-line bill payment, and on-line documentation and data exchange. In addition, the architecture of servers providing services to be accessed over a network often consists of many different servers around the world instead of just one. Although they currently enjoy the comfort and convenience of the internet, people are facing emerging challenges with regard to network security.

Identity authentication is the key security issue facing various types of on-line applications and service systems. Before a user accesses services provided by a service provider server, mutual identity authentication between the user and server is needed to prevent unauthorized

Science and Technology Program (No. J16LN61 to X.W.).

Competing interests: The authors have declared that no competing interests exist.

personnel from accessing services provided by the server and avoiding an illegal system defrauding the user by masquerading as a legitimate server. In a single-server environment, password-based authentication schemes [1] and enhanced versions that additionally use smart cards [2–9] are widely used to provide mutual authentication between the users and servers. However, conventional password-based authentication methods are not suitable for multi-server environments since each user need to not only log into various remote servers repetitively but also remember many different sets of identities and passwords if he/she wants to access these service provider servers. To resolve this problem, in 2000, based on the difficulty of factorization and hash functions, Lee and Chang [10] proposed a user identification and key distribution scheme that can be applied to multi-server environments. Since then, authentication schemes for multi-server environments have been widely investigated and designed by many researchers [11–37].

Based on the utilized basic cryptographic algorithms, multi-server authentication schemes can be divided into two types: hash-based authentication schemes and public-key-based authentication schemes. Simultaneously, among existing multi-server authentication schemes, some of them need a registration center (RC) to participate in the authentication and session key agreement phase, whereas others do not have this requirement. Therefore, based on whether the RC participates in the authentication and session key agreement phase, we divide the multi-server authentication schemes into RC-dependent authentication schemes and non-RC-dependent authentication schemes.

In this paper, we analyze a novel multi-server authentication scheme, Li et al.'s scheme [20], which is only based on hash functions and a non-RC-dependent authentication scheme. We find that this scheme is vulnerable to stolen smart cards and offline dictionary attacks, replay attacks, impersonation attacks and server spoofing attacks. By analyzing other similar schemes [15, 17–19], we find that the type of dynamic ID-based multi-server authentication scheme that only uses hash functions and are not dependent on RCs face difficulties in providing perfectly efficient and secure authentication. To compensate for these shortcomings, we propose a novel dynamic ID-based remote user authentication scheme for multi-server environments. Compared with previous related works, our scheme has many advantages. First, the scheme enjoys important security attributes, including being able to prevent various attacks, user anonymity, a lack of verification table, and local password verification. Second, the scheme does not use a timestamp; therefore, it avoids the clock synchronization problem. Further, the scheme uses self-certified public keys, by which the user's public key can be computed directly from the signature of the trusted third party on the user's identity instead of verifying the public key using an explicit signature on a user's public key. Therefore, our scheme is more practical and universal for multi-server environments. Finally, the performance and cost analysis show that our scheme is very efficient and more secure than other related schemes.

Related works

A large number of authentication schemes have been proposed for multi-server environments. Hash functions are a key technology in the construction of multi-server authentication schemes. In 2004, Juang et al. [11] proposed an efficient multi-server password authenticated key agreement scheme based on a hash function and symmetric key cryptosystem. In 2009, Hsiang and Shih [12] proposed a dynamic ID-based remote user authentication scheme for multi-server environments in which only a hash function is used. However, Sood et al. [13] found that Hsiang and Shih's scheme is susceptible to replay attacks, impersonation attacks and stolen smart card attacks. Moreover, the password change phase of Hsiang and Shih's scheme is insecure. Later, Sood et al. presented a novel dynamic identity-based authentication

protocol for multi-server architectures to resolve the security flaws of Hsiang and Shih's scheme [13]. In addition, Sood et al.'s protocol is practical and computationally efficient because only nonce, one-way hash functions and XOR operations are used in its implementation. After that, Li et al. [14] noted that Sood et al.'s protocol remains vulnerable to leak-of-verifier attacks, stolen smart card attacks and impersonation attacks. Simultaneously, Li et al. [14] proposed another dynamic identity-based authentication protocol for multi-server architectures. However, the above-mentioned schemes are all RC-dependent multi-server authentication schemes. In 2009, Liao and Wang [15] proposed a dynamic ID-based multi-server authentication scheme that is based on hash functions and does not depend on RCs. This scheme not only satisfies all requirements for multi-server environments but also achieves efficient computation. However, Liao and Wang's scheme has been found to be vulnerable to insider attacks, masquerade attacks, server spoofing attacks, and registration center spoofing attacks and is not reparable [16]. Later, Shao et al. [17] and Lee et al. [18, 19] proposed similar types of multi-server authentication schemes. In 2012, Li et al. [20] noted that Lee et al.'s scheme [18] cannot withstand forgery attacks or server spoofing attacks and cannot provide proper authentication; they then proposed a novel dynamic ID-based multi-server authentication scheme that only uses a hash function and is not dependent on RCs. Moreover, the scheme is found to be suitable for financial security authentication. However, through careful analysis, we find that Li et al.'s scheme [20] remains vulnerable to stolen smart card and offline dictionary attacks, replay attacks, impersonation attacks and server spoofing attacks. We also analyzed Shao et al.'s scheme [17] and Lee et al.'s scheme [19]; they are all vulnerable to stolen smart card and offline dictionary attacks, replay attacks, impersonation attacks and server spoofing attacks. In general, it is difficult to construct a secure dynamic ID-based and non-RC-dependent multi-server authentication scheme if only hash functions are used.

Public-key cryptography is another useful technique that is widely used in the construction of multi-server authentication schemes. In 2000, Lee and Chang [21] proposed a user identification and key distribution scheme in which the difficulty of factorization on public key cryptography is used. In 2001, Tsaur [22] proposed a remote user authentication scheme based on an RSA cryptosystem and Lagrange interpolating polynomials for multi-server environments. Then, Lin et al. [23] proposed a multi-server authentication protocol based on the simple geometric properties of the Euclidean and discrete logarithm problem concept. In their scheme, the system does not need to maintain a verification table, and the users who have registered with the servers do not need to remember different login passwords for various servers. Since traditional public key cryptographic algorithms require many expensive computations and consume substantial energy, Geng and Zhang [24] proposed a dynamic ID-based user authentication and key agreement scheme for multi-server environments using bilinear pairings. However, Geng and Zhang's scheme cannot withstand user spoofing attacks [25]. Later, Tseng et al. [26] proposed an efficient pairing-based user authentication scheme with smart cards. Performance analysis and experimental data demonstrate that their scheme is well suited for mobile devices with limited computing capabilities. However, in 2013, Liao and Hsiao [27] noted that Tseng et al.'s scheme is vulnerable to insider attacks, offline dictionary attacks and malicious server attacks and cannot provide proper mutual authentication and session key agreement. Simultaneously, Liao and Hsiao proposed a novel non-RC-dependent multi-server remote user authentication scheme using self-certified public keys for mobile clients [27]. Recently, Chou et al. [28] found that Liao and Hsiao's scheme cannot withstand password guessing attacks. Furthermore, through careful analysis, we found that Liao and Hsiao's scheme remains vulnerable to denial of service attacks and cannot ensure a user's anonymity or provide local password verification. In this paper, we propose a secure dynamic ID-based

and non-RC-dependent multi-server authentication scheme using pairing and self-certified public keys.

Preliminaries

In this section, we introduce the concepts of bilinear pairings, self-certified public keys, as well as some related mathematical assumptions.

Bilinear pairings

Let G_1 be an additive cyclic group with a large prime order q, and let G_2 be a multiplicative cyclic group with the same order q. In particular, G_1 is a subgroup of the group of points on an elliptic curve over a finite field $E(F_p)$, and G_2 is a subgroup of the multiplicative group over a finite field. P is a generator of G_1.

A bilinear pairing is a map e: $G_1 \times G_1 \rightarrow G_2$ and satisfies the following properties:

(1) Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_q^*$.

(2) Non-degenerate: There exists $P, Q \in G_1$ such that $e(P, Q) \neq 1$.

(3) Computability: There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

Self-certified public keys

In [27], Liao et al. first proposed a key distribution scheme based on self-certified public keys (SCPKs) [38, 39] among the service servers. Using the SCPK, a user's public key can be computed directly from the signature of the trusted third party (TTP) on the user's identity instead of verifying the public key using an explicit signature on a user's public key. The SCPK scheme is described as follows.

(1) Initialization: The trusted third party (TTP) first generates all the needed parameters of the scheme. The TTP chooses a non-singular high elliptic curve $E(F_p)$ defined over a finite field, which is used with a point-based generator P of prime order q. Then, the TTP freely chooses his/her secret key s_T and computes his/her public key $pub_T = s_T \cdot P$. The related parameters and pub_T are publicly and authentically available.

(2) Private key generation: A user A chooses a random number k_A, computes $K_A = k_A \cdot P$ and sends his/her identity ID_A and K_A to the TTP. The TTP chooses a random number r_A, computes $W_A = K_A + r_A \cdot P$ and $\bar{s}_A = s_T \cdot h(ID_A \| W_A) + r_A$, and sends W_A and $\bar{s}_A$ to user A. Then, A obtains his/her secret key by calculating $s_A = \bar{s}_A + k_A$.

(3) Public key extraction: Anyone can calculate A's public key $pub_A = h(ID_A \| W_A)pub_T + W_A$ given W_A.

Related mathematical assumptions

To prove the security of our proposed protocol, we present some important mathematical problems and assumptions for bilinear pairings defined on elliptic curves. The related concrete description can be found in [40, 41].

(1) Computational discrete logarithm (CDL) problem: Given $R = x \cdot P$, where $P, R \in G_1$, it is easy to calculate R given x and P, but it is hard to determine x given P and R.

(2) Elliptic curve factorization (ECF) problem: Given two points P and $R = x \cdot P + y \cdot P$ for $x, y \in Z_q^*$, it is hard to find $x \cdot P$ and $y \cdot P$.

(3) Computational Diffie-Hellman (CDH) problem: Given $P, xP, yP \in G_1$, it is hard to compute $xyP \in G_1$.

Review and cryptanalysis of Li et al.'s authentication scheme

Review of Li et al.'s scheme

There are three participants in Li et al.'s scheme: the registration center RC, the server S_j, and the user U_i. RC generates the master secret key x and a secret number y to construct $h(x||y)$ and $h(SID_j||h(y))$, in which SID_j is the identity of server S_j; then, it delivers them to the server S_j through a secure channel. Li et al.'s scheme contains four phases:the registration phase, the login phase, the verification phase and the password change phase.

Registration phase. When the remote user authentication scheme starts, the registration process should be first performed by the user U_i and RC:

(1) U_i generates a random number b and freely chooses his/her identity ID_i and the password PW_i. Then, U_i calculates $A_i = h(b \oplus PW_i)$. After that, U_i transmits ID_i and A_i to RC for registration through a secure channel.

(2) RC computes $B_i = h(ID_i||x)$, $C_i = h(ID_i||h(y)||A_i)$, $D_i = h(B_i||h(x||y))$ and $E_i = B_i \oplus h(x||y)$. Then, RC stores $\{C_i, D_i, E_i, h(\cdot), h(y)\}$ on the smart card of U_i and sends it to U_i by a secure channel.

(3) U_i adds the random number b into the smart card, which ultimately possesses the information $\{C_i, D_i, E_i, b, h(\cdot), h(y)\}$.

Login phase. When user U_i wants to log into the server S_j, the following procedures should be performed:

(1) After the smart card is inserted into the card reader, the user is prompted to enter his/her ID_i and PW_i. After that, the smart card calculates $A_i = h(b \oplus PW_i)$, $C_i^* = h(ID_i||h(y)||A_i)$ and checks whether C_i^* is equal to C_i. If C_i^* is equal to C_i, the Login process continues. Otherwise, the session will be aborted.

(2) The smart card produces a number N_i randomly and calculates $P_{ij} = E_i \oplus h(h(SID_j||h(y))||N_i)$, $CID_i = A_i \oplus h(D_i||SID_j||N_i)$, $M_1 = h(P_{ij}||CID_i||D_i||N_i)$ and $M_2 = h(SID_j||h(y)) \oplus N_i$.

(3) The smart card transmits the login request message $\{P_{ij}, CID_i, M_1, M_2\}$ to S_j.

Verification phase. When S_j receives the login request message, the mutual authentication and session key agreement between S_j and U_i will be performed in accordance with the following steps.

(1) The server S_j calculates $N_i = M_2 \oplus h(SID_j||h(y))$, $E_i = P_{ij} \oplus h(h(SID_j||h(y))||N_i)$, $B_i = E_i \oplus h(x||y)$, $D_i = h(B_i||h(x||y))$, and $A_i = CID_i \oplus h(D_i||SID_j||N_i)$.

(2) The server S_j calculates $h(P_{ij}||CID_i||D_i||N_i)$; if the calculated result is not equal to M_1, S_j rejects the login request and aborts this session. Otherwise, S_j accepts the login request message. Then, S_j chooses a random number N_j and calculates $M_3 = h(D_i||A_i||N_j||SID_j)$, $M_4 = A_i \oplus N_i \oplus N_j$. Finally, S_j sends $\{M_3, M_4\}$ to U_i.

(3) According to the received message $\{M_3, M_4\}$, U_i calculates $N_j = A_i \oplus N_i \oplus M_4$, $M_3^* = h(D_i||A_i||N_j||SID_j)$ and verifies whether M_3^* is equal to M_3. If they are not equal, U_i rejects these messages and terminates this session. Otherwise, U_i successfully authenticates S_j. In addition, U_i calculates $M_5 = h(D_i||A_i||N_i||SID_j)$ and sends it to S_j.

(4) The server S_j computes $h(D_i||A_i||N_i||SID_j)$ and compares it with the received $\{M_5\}$ sent from U_i. If they are equal, U_i is successfully authenticated by S_j, and the mutual authentication is completed. After the mutual authentication phase, the user U_i and the server S_j calculate $SK = h(D_i||A_i||N_i||N_j||SID_j)$ as their session key in future secure communication.

Password change phase. For security, the password of the user should be changed frequently. The password change phase is performed when user U_i wants to replace the old password PW_i with a new password PW_i^{new}.

(1) The user U_i inserts his/her smart card into the card reader and inputs his/her ID_i and PW_i.

(2) The smart card calculates $A_i = h(b \oplus PW_i)$, $C_i^* = h(ID_i||h(y)||A_i)$ and verifies whether C_i^* is equal to C_i. If they are not equal, the password change request will be rejected. Otherwise, the user U_i provides a new random number b^{new} and a new password PW_i^{new}.

(3) The smart card calculates $A_i^{new} = h(b^{new} \oplus PW_i^{new})$ and $C_i^{new} = h(ID_i||h(y)||A_i^{new})$.

(4) The smart card uses C_i^{new} and b^{new} to replace C_i and b. The password change phase is completed.

Cryptanalysis of Li et al.'s scheme

Li et al. claimed that their scheme can resist many types of attacks and satisfy all the essential requirements for multi-server architecture authentication. However, if we assume that A is an adversary who has broken a user U_m and a server S_n or a combination of a malicious user U_m and a dishonest server S_n, then A can obtain the secret number $h(x||y)$ and $h(y)$ and perform stolen smart card and offline dictionary attacks, replay attacks, impersonation attacks and server spoofing attacks on Li et al.'s scheme. The concrete cryptanalysis of the Li et al.'s scheme is shown as follows.

Stolen smart card and offline dictionary attacks. If a user U_i's smart card is stolen by an adversary A, A can extract the information $\{C_i, D_i, E_i, b, h(\cdot), h(y)\}$ from the memory of the stolen smart card. Furthermore, if A intercepts a valid login request message $\{P_{ij}, CID_i, M_1, M_2\}$ sent from user U_i to server S_j in the public communication channel, A can compute $N_i = h(SID_j||h(y)) \oplus M_2$, $E_i = P_{ij} \oplus h(h(SID_j||h(y))||N_i)$, $B_i = E_i \oplus h(x||y)$, $D_i = h(B_i||h(x||y))$ and $A_i = CID_i \oplus h(D_i||SID_j||N_i)$ using $h(y)$ and $h(x||y)$. Then, A can launch an offline dictionary attack on $C_i = h(ID_i||h(y)||A_i)$ to determine the identity ID_i of user U_i because A knows the values of A_i and $h(y)$ corresponding to the user U_i. In addition, A can launch offline dictionary attacks on $A_i = h(b \oplus PW_i)$ to determine the password PW_i of U_i because A knows the value of b from the stolen smart card of the user U_i. Now, A possesses the valid smart card of user U_i, knows the identity ID_i and password PW_i corresponding to user U_i and hence can login to any service provider server.

Replay attacks. A replay attack is when an adversary replays the same message of a receiver or sender again. If adversary A has intercepted a valid login request message $\{P_{ij}, CID_i, M_1, M_2\}$ sent from user U_i to server S_j in the public communication channel, then A can compute $N_i = h(SID_j||h(y)) \oplus M_2$, $E_i = P_{ij} \oplus h(h(SID_j||h(y))||N_i)$, $B_i = E_i \oplus h(x||y)$, $D_i = h(B_i||h(x||y))$ and $A_i = CID_i \oplus h(D_i||SID_j||N_i)$ using $h(y)$ and $h(x||y)$. Then, adversary A can replay this login request message $\{P_{ij}, CID_i, M_1, M_2\}$ to S_j by masquerading as the user U_i at some later time. After verification of the login request message, S_j computes $M_3 = h(D_i||A_i||N_j||SID_j)$ and $M_4 = A_i \oplus N_i \oplus N_j$ and sends the message $\{M_3, M_4\}$ to A, who is masquerading as the user U_i. The adversary A can verify the received value of $\{M_3, M_4\}$ and compute $M_5' = h(D_i||A_i||N_i||SID_j)$ since they know the values of N_i, E_i, B_i, D_i and A_i. Then, A sends $\{M_5'\}$ to the server S_j. The server S_j computes $h(D_i||A_i||N_i||SID_j)$ and checks it with the received message $\{M_5'\}$. This equivalency authenticates the legitimacy of the user U_i and the service provider server S_j, and the login request is accepted. Finally, after mutual authentication, adversary A masquerading as the user U_i and the server S_j agree on the common session key as $SK = h(D_i||A_i||N_i||N_j||SID_j)$. Therefore, the adversary A can masquerade as user U_i to login to server S_j by replaying the same login request message that had been sent from U_i to S_j.

Impersonation attacks. In this subsection, we show that an adversary A who possesses $h(y)$ and $h(x||y)$ can masquerade as any user U_i to login to any server S_j as follows.

Adversary A chooses two random numbers a_i and b_i and computes $A_i = h(a_i)$ and $B_i = h(b_i)$. Then, A can compute $D_i = h(B_i||h(x||y))$, $E_i = B_i \oplus h(x||y)$, $P_{ij} = E_i \oplus h(h(SID_j||h(y))||N_i)$, $CID_i = A_i \oplus h(D_i||SID_j||N_i)$, $M_1 = h(P_{ij}||CID_i||D_i||N_i)$ and $M_2 = h(SID_j||h(y)) \oplus N_i$ using $h(y)$ and $h(x||y)$. Now, A sends the login request message $\{P_{ij}, CID_i, M_1, M_2\}$ by masquerading as the user U_i to server S_j. After receiving the login request message, S_j computes $N_i = h(SID_j||h(y)) \oplus M_2$, $E_i = P_{ij} \oplus h(h(SID_j||h(y))||N_i)$, $B_i = E_i \oplus h(x||y)$, $D_i = h(B_i||h(x||y))$ and $A_i = CID_i \oplus h(D_i||SID_j||N_i)$ using $\{P_{ij}, CID_i, M_1, M_2\}$, $h(x||y)$ and $h(SID_j||h(y))$. Then, S_j computes $M_3 = h(D_i||A_i||N_j||SID_j)$ and $M_4 = A_i \oplus N_i \oplus N_j$ and sends the message $\{M_3, M_4\}$ to A, who is masquerading as the user U_i. Then, adversary A computes $N_j = A_i \oplus N_i \oplus M_4$ and verifies M_3 by computing $h(D_i||A_i||N_j||SID_j)$. Then, A computes $M_5 = h(D_i||A_i||N_i||SID_j)$ and sends $\{M_5\}$ back to the server S_j. The server S_j computes $h(D_i||A_i||N_i||SID_j)$ and checks it against the received message $\{M_5\}$. This equivalency authenticates the legitimacy of the user U_i and the service provider server S_j, and the login request is accepted. Finally, after mutual authentication, adversary A masquerading as the user U_i and the server S_j agree on the common session key as $SK = h(D_i||A_i||N_i||N_j||SID_j)$.

Server spoofing attacks. In this subsection, we show that an adversary A who possesses $h(y)$ and $h(x||y)$ can masquerade as the server S_j to spoof user U_i if A has intercepted a valid login request message $\{P_{ij}, CID_i, M_1, M_2\}$ sent from user U_i to server S_j over a public communication channel.

After intercepting a valid login request message $\{P_{ij}, CID_i, M_1, M_2\}$ sent from user U_i to server S_j over a public communication channel, A can compute $N_i = h(SID_j||h(y)) \oplus M_2$, $E_i = P_{ij} \oplus h(h(SID_j||h(y))||N_i)$, $B_i = E_i \oplus h(x||y)$, $D_i = h(B_i||h(x||y))$ and $A_i = CID_i \oplus h(D_i||SID_j||N_i)$ corresponding to U_i. Then, A can choose a random number N_j' and compute $M_3 = h(D_i||A_i||N_j'||SID_j)$ and $M_4 = A_i \oplus N_i \oplus N_j'$. A then sends the message $\{M_3, M_4\}$ by masquerading as the server S_j to the user U_i. After receiving the message $\{M_3, M_4\}$, U_i computes $N_j' = A_i \oplus N_i \oplus M_4$ and verifies M_3 by computing $h(D_i||A_i||N_j'||SID_j)$. Then, U_i computes $M_5 = h(D_i||A_i||N_i||SID_j)$ and sends it to the server S_j, who is masquerading as the adversary A. Then, A computes $h(D_i||A_i||N_i||SID_j)$ and checks it against the received message $\{M_5\}$. Finally, after mutual authentication, the adversary A masquerading as the server S_j and the user U_i agree on the common session key as $SK = h(D_i||A_i||N_i||N_j'||SID_j)$.

Discussion

Except for Li et al.'s scheme, we also analyzed four other dynamic ID-based authentication schemes for multi-server environments [15, 17–19]. These schemes are all based on hash functions and are not dependent on RCs. We found that this type of multi-server remote user authentication scheme is generally vulnerable to stolen smart card and offline dictionary attacks, impersonation attacks, server spoofing attacks etc. The cryptanalysis methods used by these schemes are similar to that of Li et al.'s scheme shown in Section 4.2. We believe that under the assumptions that no RC participates in the authentication and session key agreement phase, the dynamic ID and hash function-based user authentication schemes for multi-server environments face difficulties in providing perfectly efficient and secure authentication. Fortunately, there is another technique, public-key cryptography, that is widely used in the construction of authentication schemes. Therefore, to construct a secure, low-power-consumption and non-RC-dependent authentication scheme, we adopt the elliptic curve cryptographic technology of public-key techniques, and we propose a novel dynamic ID-based and non-RC-dependent remote user authentication scheme using pairing and self-certified public keys for multi-server environments.

Table 1. Notations used in the proposed scheme.

e	A bilinear map, e: $G_1 \times G_1 \rightarrow G_2$.
U_i	The ith user.
ID_i	The identity of the user U_i.
S_j	The jth service provider server.
SID_j	The identity of the service provider server S_j.
RC	The registration center.
s_{RC}	The master secret key of the registration center RC in Z_q^*.
pub_{RC}	The public key of RC, $pub_{RC} = s_{RC} \cdot P$.
P	A generator of group G_1.
$H()$	A map-to-point function, H: 0, $1^* \rightarrow G_1$.
$h()$	A one-way hash function, h: 0, $1^* \rightarrow 0$, 1^k, where k is the output length. $h()$ allows the concatenation of some integer values and points on an elliptic curve.
$\oplus$	A simple XOR operation in G_1. If P_1, $P_2 \in G_1$, P_1 and P_2 are points on an elliptic curve over a finite field, the operation $P_1 \oplus P_2$ means that it performs the XOR operations of the x-coordinates and y-coordinates of P_1 and P_2, respectively.
$\|$	The concatenation operation.

https://doi.org/10.1371/journal.pone.0202657.t001

The proposed scheme

In this section, we propose a novel dynamic ID-based and non-RC-dependent remote user authentication scheme for multi-server environments using pairing and self-certified public keys. Our scheme contains three participants: the user U_i, the service provider server S_j, and the registration center RC. A legitimate user U_i can easily login to the service provider server using his smart card, identity and password. There are six phases in the proposed scheme: the system initialization phase, the user registration phase, the server registration phase, the login phase, the authentication and session key agreement phase, and the password change phase. The notations used in our proposed scheme are summarized in Table 1.

System initialization phase

In the proposed scheme, the registration center RC is assumed to be a TTP. In the system initialization phase, RC generates all the needed parameters of the scheme.

(1) The RC selects a cyclic additive group G_1 of prime order q, a cyclic multiplicative group G_2 of the same order q, a generator P of G_1, and a bilinear map e: $G_1 \times G_1 \rightarrow G_2$.

(2) The RC freely chooses a number $s_{RC} \in Z_q^*$ held as the system private key and computes $pub_{RC} = s_{RC} \cdot P$ as the system public key.

(3) The RC selects two cryptographic hash functions $H(\cdot)$ and $h(\cdot)$.

Finally, all the related parameters $\{e, G_1, G_2, q, P, Pub_{RC}, H(\cdot), h(\cdot)\}$ are publicly and authentically available.

User registration phase

When the user U_i wants to access the services, he/she has to submit some of his/her related information to the registration center RC for registration. The steps of the user registration phase are as follows:

(1) U_i freely generates his/her identity ID_i and password pw_i and chooses a random number b_i. Then, U_i computes $HPW_i = h(ID_i \| pw_i \| b_i) \cdot P$ and submits ID_i and HPW_i to RC for registration through a secure channel.

(2) When receiving the message ID_i and HPW_i, RC computes $QID_i = H(ID_i)$, $CID_i = s_{RC} \cdot QID_i$, $Reg_{ID_i} = CID_i \oplus s_{RC} \cdot HPW_i$ and $H_i = h(QID_i \| CID_i)$. Then, RC stores the message $\{Reg_{ID_i}, H_i\}$ in U_i's smart card and submits the smart card to U_i through a secure channel.

(3) After receiving the smart card, U_i enters b_i into the smart card. Finally, the smart card contains the parameters $\{Reg_{ID_i}, H_i, b_i\}$.

Server registration phase

If a service provider server S_j wants to provide services to the users, he/she must perform the registration to the registration center RC to become a legal service provider server. The process of the server registration phase of the proposed scheme is based on SCPK.

(1) S_j chooses a random number v_j and computes $V_j = v_j \cdot P$. Then, S_j submits SID_j and V_j to RC for registration via a secure channel.

(2) After receiving the message $\{SID_j, V_j\}$, RC chooses a random number w_j and computes $W_j = w_j \cdot P + V_j$ and $s'_j = (s_{RC} \cdot h(SID_j \| W_j) + w_j) \bmod q$. Then, RC submits the message $\{W_j, s'_j\}$ to S_j through a secure channel.

(3) After receiving $\{W_j, s'_j\}$, S_j computes their private key $s_j = (s'_j + v_j) \bmod q$ and checks the validity of the values issued to them by checking the following equation: $pub_j = s_j \cdot P = h(SID_j \| W_j) \cdot pub_{RC} + W_j$. Finally, S_j's personal information contains $\{SID_j, pub_j, s_j, W_j\}$

The details of the user registration phase and server registration phase are shown in Fig 1.

Login phase

If user U_i wants to access the services provided by server S_j, U_i needs to login to S_j, where the process of the login phase are as follows:

(1) The user U_i inserts their smart card into the smart card reader and inputs their identity ID_i and password pw_i. The smart card then calculates $QID_i = H(ID_i)$, $CID_i = Reg_{ID_i} \oplus h(ID_i \| pw_i \| b_i) \cdot pub_{RC}$, and $H_i^* = h(QID_i \| CID_i)$ and verifies whether H_i^* is equal to H_i. If they are equal, it is verified that U_i has the correct user identity and password. Thus, U_i is a legitimate user. Otherwise, the smart card aborts the session.

(2) The smart card chooses two random numbers u_i and r_i, and it computes $DID_i = u_i \cdot QID_i$ and $R_i = r_i \cdot P$. Then, the smart card sends the login request message $\{DID_i, R_i\}$ to server S_j over a public channel.

Authentication and session key agreement phase

(1) Based on the received login request message $\{DID_i, R_i\}$ sent from the user U_i, the server S_j chooses a random number r_j and computes $R_j = r_j \cdot P$, $T_{ji} = r_j \cdot R_i$, $K_{ji} = s_j \cdot R_i$ and $Auth_{ji} = h(DID_i \| SID_j \| K_{ji} \| R_j)$. Then, S_j sends the message $\{W_j, R_j, Auth_{ji}\}$ to U_i.

(2) When receiving $\{W_j, R_j, Auth_{ji}\}$, U_i computes $T_{ij} = r_i \cdot R_j$, $pub_j = h(SID_j \| W_j) \cdot pub_{RC} + W_j$, $K_{ij} = r_i \cdot pub_j$ and $Auth_{ij} = h(DID_i \| SID_j \| K_{ij} \| R_j)$. Then, U_i checks $Auth_{ij}$ with the received $Auth_{ji}$. If they are not equal, U_i terminates this session. Otherwise, S_j is proven to have the correct private key s_j, and thus, S_j is authenticated. U_i continues to compute $M_i = r_i \cdot DID_i$, $N_i = u_i \cdot CID_i$, $d_{ij} = h(DID_i \| SID_j \| K_{ij} \| M_i)$ and $B_i = (r_i + d_{ij}) \cdot N_i$. Finally, U_i sends the message $\{M_i, B_i\}$ to S_j.

(3) After receiving the message $\{M_i, B_i\}$ sent from U_i, S_j computes $d_{ji} = h(DID_i \| SID_j \| K_{ji} \| M_i)$ and checks whether $e(M_i + d_{ji} \cdot DID_i, pub_{RC}) = e(B_i, P)$. If they are not equal, S_j terminates this session. Otherwise, U_i is authenticated.

Finally, the user U_i and the server S_j agree on a common session key as U_i: $SK = h(DID_i \| SID_j \| K_{ij} \| T_{ij})$, S_j: $SK = h(DID_i \| SID_j \| K_{ji} \| T_{ji})$.

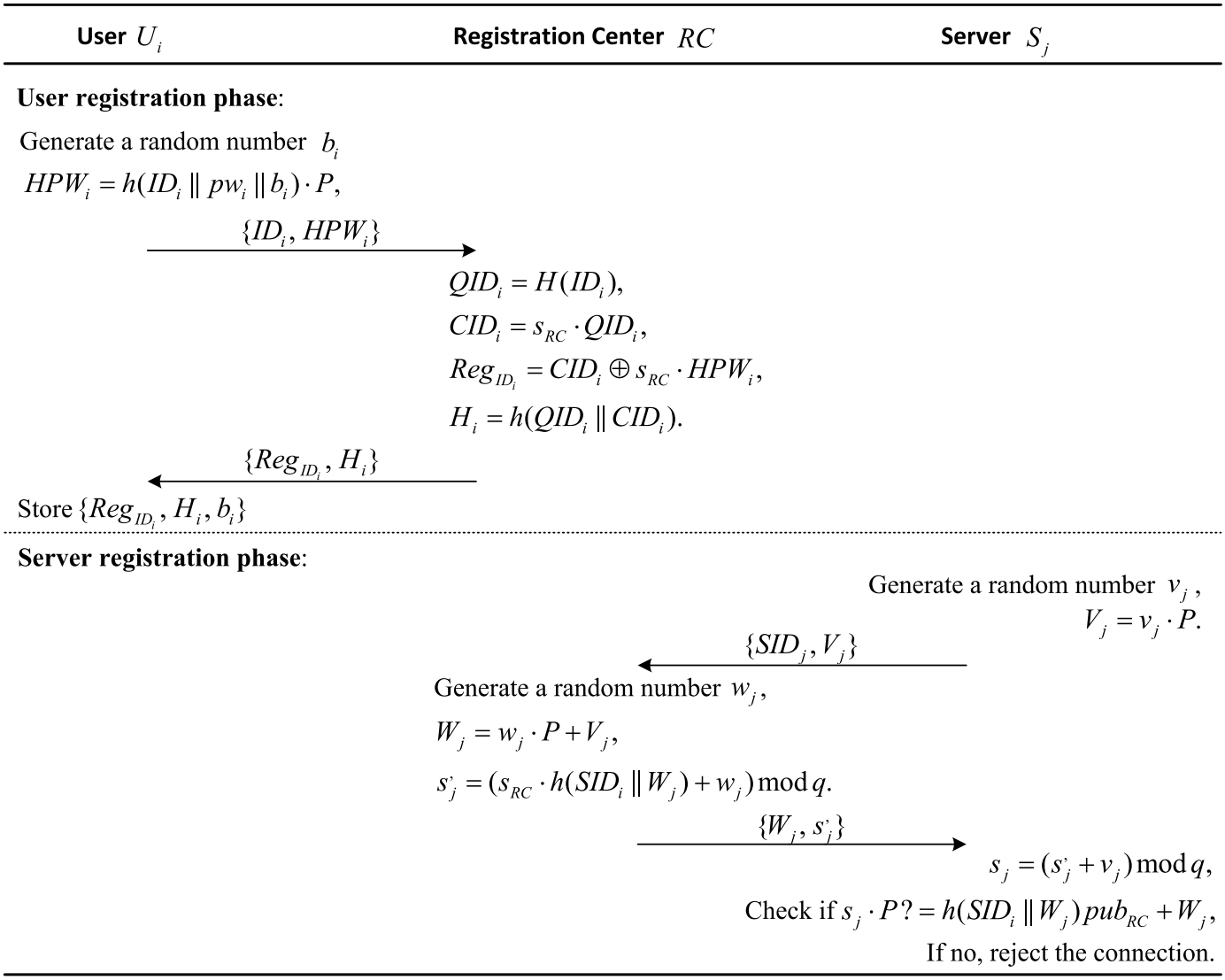

Fig 1. User and server registration phases of the proposed scheme.

https://doi.org/10.1371/journal.pone.0202657.g001

Sections 5.4 and 5.5 give the detailed procedures of the login phase and authentication and session key agreement phase, which are also depicted in Fig 2.

Password change phase

For security purposes, users need to change their passwords frequently. The following steps show the password change phase process for a user U_i.

(1) The user U_i inserts his/her smart card into the smart card reader and inputs their identity ID_i and password pw_i. Then, the smart card computes $QID_i = H(ID_i)$, $CID_i = Reg_{ID_i} \oplus h(ID_i \| pw_i \| b_i) \cdot pub_{RC}$, $H_i^* = h(QID_i \| CID_i)$ and checks whether $H_i^* = H_i$. If they are equal, U_i is verified as a legitimate user; otherwise, the smart card rejects the password change request.

User U_i		Server S_j
Login and verification phase:		
Insert smart card, and input ID_i, pw_i,		
$QID_i = H(ID_i)$,		
$CID_i = Reg_{ID_i} \oplus h(ID_i \| pw_i \| b_i) \cdot pub_{RC}$,		
Check if $H_i^* = h(QID_i \| CID_i)? = H_i$,		
If no, reject the login request,		
Generate a random number u_i, r_i,		
$DID_i = u_i \cdot QID_i$,		
$R_i = r_i \cdot P$.	$\{DID_i, R_i\}$ →	
		Generate a random number r_j,
		$R_j = r_j \cdot P$,
		$T_{ji} = r_j \cdot R_i$,
		$K_{ji} = s_j \cdot R_i$,
		$Auth_{ji} = h(DID_i \| SID_j \| K_{ji} \| R_j)$.
	← $\{W_j, R_j, Auth_{ji}\}$	
$T_{ij} = r_i \cdot R_j$,		
$pub_j = h(SID_j \| W_j) \cdot pub_{RC} + W_j$,		
$K_{ij} = r_i \cdot pub_j$,		
Check if $Auth_{ij} = h(DID_i \| SID_j \| K_{ij} \| R_j)? = Auth_{ji}$,		
If yes, S_j is authenticated,		
$M_i = r_i \cdot DID_i$,		
$N_i = u_i \cdot CID_i$,		
$d_{ij} = h(DID_i \| SID_j \| K_{ij} \| M_i)$,		
$B_i = (r_i + d_{ij}) \cdot N_i$.	$\{M_i, B_i\}$ →	
		$d_{ji} = h(DID_i \| SID_j \| K_{ji} \| M_i)$,
		Check if $e(M_i = d_{ji} \cdot DID_i, pub_{RC})? + e(B_i, P)$,
		If yes, U_i is authenticated.
$SK = h(DID_i \| SID_j \| K_{ij} \| T_{ij})$		$SK = h(DID_i \| SID_j \| K_{ji} \| T_{ji})$

Fig 2. Login phase and authentication and session key agreement phase.

https://doi.org/10.1371/journal.pone.0202657.g002

(2) The smart card generates a random number z_i and computes $Z_i = z_i \cdot P$ and $AID_i = CID_i \oplus z_i \cdot pub_{RC}$. Then, the smart card sends the message $\{ID_i, AID_i, Z_i\}$ to the registration center RC.

(3) After receiving the message $\{ID_i, AID_i, Z_i\}$, RC computes $CID_i = AID_i \oplus s_{RC} \cdot Z_i$, $QID_i = H(ID_i)$, and checks whether $e(CID_i, P) = e(QID_i, pub_{RC})$. If they are equal, user U_i is authenticated. Then, RC computes $V_1 = h(CID_i \parallel s_{RC} \cdot Z_i)$ and sends $\{V_1\}$ to U_i.

(4) When receiving $\{V_1\}$, the user computes $V_1^* = h(CID_i \| z_i \cdot pub_{RC})$ and checks it against the received V_1. If they are equal, the registration center RC is authenticated. Then, U_i chooses his/her new password pw_i^{new} and the new random number b_i^{new}, and they compute $HPW_i^{new} = h(ID_i \| pw_i^{new} \| b_i^{new}) \cdot P$, $V_2 = HPW_i^{new} \oplus z_i \cdot pub_{RC}$ and $V_3 = h(CID_i \| z_i \cdot pub_{RC} \| HPW_i^{new})$. Then, U_i submits $\{V_2, V_3\}$ to RC.

(5) Upon receiving the response $\{V_2, V_3\}$, the registration server RC computes $HPW_i^{new} = V_2 \oplus s_{RC} \cdot Z_i$ and $V_3^* = h(CID_i \| s_{RC} \cdot Z_i \| HPW_i^{new})$. Then, RC compares V_3^* with the received V_3. If they are equal, RC continues to compute $Reg_{ID_i}^{new} = CID_i \oplus s_{RC} \cdot HPW_i^{new}$, $V_4 = Reg_{ID_i}^{new} \oplus s_{RC} \cdot Z_i$ and $V_5 = h(s_{RC} \cdot Z_i \| Reg_{ID_i}^{new})$. After that, RC sends $\{V_4, V_5\}$ to U_i.

(6) After receiving $\{V_4, V_5\}$, U_i computes $Reg_{ID_i}^{new} = V_4 \oplus z_i \cdot pub_{RC}$ and $V_5^* = h(z_i \cdot pub_{RC} \| Reg_{ID_i}^{new})$. Then, U_i checks whether $V_5^* = V_5$. If they are equal, user U_i replaces the original Reg_{ID_i} and b_i with $Reg_{ID_i}^{new}$ and b_i^{new}.

In addition to the descriptions listed above, the procedures of the password change phase of the proposed scheme are also given in Fig 3.

Security analysis

Stolen smart card and offline dictionary attacks

In the proposed scheme, we assume that if a smart card is stolen, physical protection methods cannot prevent malicious attackers for obtaining the stored secure elements. Simultaneously, an adversary A can access a large dictionary of words that likely includes the user's password and intercept the communications between the user and server.

In the proposed scheme, if a user U_i's smart card is stolen by an adversary A, the latter can extract $\{Reg_{ID_i}, H_i, b_i\}$ from the memory of the stolen smart card. Simultaneously, it is assumed that adversary A has intercepted a previous full session of messages $\{DID_i, R_i, W_j, R_j, Auth_{ji}, M_i, B_i\}$ between the user U_i and server S_j. However, the adversary still cannot obtain U_i's identity ID_i and password pw_i except by guessing ID_i and pw_i simultaneously. Therefore, it is impossible to obtain U_i's identity ID_i and password pw_i from a stolen smart card and using offline dictionary attacks in our proposed scheme.

Replay attacks

Replaying a message of a previous session into a new session is useless in our proposed scheme because the user's smart card and the server choose different rand numbers r_i and r_j, and the user's identity is different in each new session. These factors make all messages dynamic and valid for that session only. If we assume that an adversary A replies with an intercepted previous login request $\{DID_i, R_i\}$ to S_j, after receiving the response message $\{W_j, R_j, Auth_{ji}\}$ sent from S_j, A cannot compute the correct response message $\{M_i, B_i\}$ to pass S_j's authentication since they do not know the values of ID_i, pw_i, u_i and r_i. Therefore, the proposed scheme is robust to replay attacks.

User U_i	Registration Center RC
Password change phase:	
Insert smart card, and input ID_i, pw_i,	
$QID_i = H(ID_i)$,	
$CID_i = Reg_{ID_i} \oplus h(ID_i \| pw_i \| b_i) \cdot pub_{RC}$,	
Check if $H_i^* = h(QID_i \| CID_i)? = H_i$,	
If no, reject the request,	
Generate a random number z_i,	
$Z_i = z_i \cdot P$,	
$AID_i = CID_i \oplus z_i \cdot pub_{RC}$.	$\xrightarrow{\{ID_i, AID_i, Z_i\}}$
	$CID_i = AID_i \oplus s_{RC} \cdot Z_i$,
	$QID_i = H(ID_i)$,
	Check if $e(CID_i, P)? = e(QID_i, pub_{RC})$,
	If yes, U_i is authenticated,
	$V_1 = h(CID_i \| s_{RC} \cdot Z_i)$.
	$\xleftarrow{\{V_1\}}$
Check if $V_1^* = h(CID_i \| z_i \cdot pub_{RC})? = V_1$,	
If yes, RC is authenticated,	
$HPW_i^{new} = h(ID_i \| pw_i^{new} \| b_i^{new}) \cdot P$,	
$V_2 = HPW_i^{new} \oplus z_i \cdot pub_{RC}$,	
$V_3 = h(CID_i \| z_i \cdot pub_{RC} \| HPW_i^{new})$.	
	$\xrightarrow{\{V_2, V_3\}}$
	$HPW_i^{new} = V_2 \oplus s_{RC} \cdot Z_i$,
	Check if $V_3^* = h(CID_i \| s_{RC} \cdot Z_i \| HPW_i^{new})? = V_3$,
	If yes, $Reg_{ID_i}^{new} = CID_i \oplus s_{RC} \cdot HPW_i^{new}$,
	$V_4 = Reg_{ID_i}^{new} \oplus s_{RC} \cdot Z_i$,
	$V_5 = h(s_{RC} \cdot Z_i \| Reg_{ID_i}^{new})$.
	$\xleftarrow{\{V_4, V_5\}}$
$Reg_{ID_i}^{new} = V_4 \oplus z_i \cdot pub_{RC}$,	
Check if $V_5^* = h(z_i \cdot pub_{RC} \| Reg_{ID_i}^{new})? = V_5$,	
If yes, replace Reg_{ID_i} and b_i with $Reg_{ID_i}^{new}$ and b_i^{new}.	

Fig 3. Password change phase of the proposed scheme.

https://doi.org/10.1371/journal.pone.0202657.g003

Impersonation attacks

If an adversary A wants to masquerade as a legitimate user U_i to pass the authentication of a server S_j, the user must have the values of both QID_i and CID_i. However, QID_i and CID_i are protected by U_i's smart card, ID_i and pw_i since $QID_i = H(ID_i)$ and $CID_i = Reg_{ID_i} \oplus h(ID_i||pw_i||b_i) \cdot pub_{RC}$. Therefore, unless the adversary A can obtain the user U_i's smart card, ID_i and pw_i simultaneously, the proposed scheme is secure to impersonation attacks.

Server spoofing attacks

If an adversary A wants to masquerade as a legal server S_j to cheat a user U_i, the adversary must calculate a valid $Auth_{ji}$ that is embedded with the shared secret key $K_{ji} = s_j \cdot R_i$ to pass the authentication of U_i. However, the adversary A cannot derive the shared secret key K_{ji} without knowing the private key s_j of the server S_j. Therefore, our scheme is secure against server spoofing attacks.

Insider attacks

In the proposed scheme, the registration center RC cannot obtain U_i's password pw_i. Since in the registration phase U_i chooses a random number b_i and sends ID_i and $HPW_i = h(ID_i \,||\, pw_i \,||\, b_i) \cdot P$ to RC, RC cannot derive pw_i from HPW_i based on the CDL problem. Therefore, the proposed scheme is robust to insider attacks.

Denial of service attacks

In denial of service attacks, an adversary A updates the identity and password verification information on the smart card to some arbitrary value, and hence, legitimate users cannot login successfully in subsequent login requests to the server. In the proposed scheme, the smart card checks the validity of user U_i's identity ID_i and password pw_i before the password update procedure. An adversary can insert the stolen smart card of the user U_i into the smart card reader and must guess the identity ID_i and password pw_i corresponding to the user U_i correctly. The smart card computes $H_i^* = h(QID_i||CID_i)$ and compares it with the stored value of H_i in its memory to verify the legitimacy of the user U_i before the smart card accepts the password update request. It is not possible to guess the identity ID_i and password pw_i correctly simultaneously in real polynomial time even after obtaining the smart card of the user U_i. Therefore, the proposed scheme is secure against denial of service attacks.

Perfect forwarding secrecy

Perfect forwarding secrecy means that even if an adversary compromises all the passwords of the users, it still cannot compromise the session key. In the proposed scheme, the session key $SK = h(DID_i \,||\, SID_j \,||\, K_{ij} \,||\, T_{ij})$ $SK = h(DID_i \,||\, SID_j \,||\, K_{ij} \,||\, T_{ji})$ is generated by three single-use random numbers u_i, r_i and r_j in each session. These single-use random numbers are only held by the user U_i and the server S_j and cannot be retrieved from SK based on the security of the CDH problem. Thus, even if an adversary obtains previous session keys, it cannot compromise other session keys. Hence, the proposed scheme achieves perfect forwarding secrecy.

User anonymity

In our proposed scheme, the user U_i's login message is different in each login phase. For each login message, $DID_i = u_i \cdot H(ID_i)$ is associated with a random number u_i, which is known by U_i

alone. Therefore, no adversary can identity the real identity of the logged on user, and our scheme can ensure the user's anonymity.

No verification table

In our proposed scheme, it is obvious that the user, server and registration center do not maintain a verification table.

Local password verification

In the proposed scheme, the smart card checks the validity of user U_i's identity ID_i and password pw_i before logging into server S_j. Since the adversary cannot compute the correct CID_i without knowledge of ID_i and pw_i to satisfy the verification equation $H_i^* = H_i$, our scheme can avoid unauthorized access via local password verification.

Proper mutual authentication

In our scheme, the user first authenticates the server. U_i sends the message $\{DID_i, R_i\}$ to the server S_j to establish a connection. After receiving the response message $\{W_j, R_j, Auth_{ji}\}$ sent from S_j, U_i computes T_{ij}, pub_j, K_{ij}, and $Auth_{ij}$ and checks whether $Auth_{ij} = Auth_{ji}$. If they are equal, S_j is authenticated by U_i. Otherwise, U_i stops to login to this server. Since $Auth_{ji} = h(DID_i \| SID_j \| K_{ji} \| R_j)$ and $K_{ji} = s_j \cdot R_i$, an adversary A cannot compute the correct K_{ji} without knowledge of the value of s_j. Any fabricated message $\{W_j', R_j', Auth_{ji}'\}$ cannot pass verification. Then, U_i computes M_i, N_i, d_{ij}, and B_i and sends the message $\{M_i, B_i\}$ to S_j. After receiving the message $\{M_i, B_i\}$ sent from U_i, S_j computes d_{ji} and checks whether $e(M_i + d_{ji} \cdot DID_i, pub_{RC}) = e(B_i, P)$. If they are not equal, S_j terminates this session; otherwise, U_i is authenticated. Since $B_i = (r_i + d_{ij}) \cdot N_i$, an adversary A cannot compute the correct B_i without knowledge of the values of u_i, r_i etc. Any fabricated message $\{M_i', B_i'\}$ cannot pass verification. Therefore, our proposed scheme can provide proper mutual authentication.

Performance comparison and functionality analysis

In this section, we compare the performance and functionality of our proposed scheme with some previous schemes. To analyze the computation cost, some notations are defined as follows.

TG_e: The time for executing a bilinear map operation, e: $G_1 \times G_1 \rightarrow G_2$.

TG_{mul}: The time for executing point scalar multiplication on the group G_1.

TG_H: The time for executing a map-to-point hash function H(.).

TG_{add}: The time for executing point addition on the group G_1.

T_h: The time for executing a one-way hash function h(.).

Since the XOR operation and the modular multiplication operation require very few computations, it is usually negligible considering their computation costs.

Table 2 shows the performance comparisons of our proposed scheme and various other related protocols. We focus on three computational costs: C1, the total time for all operations executed during the user registration phase; C2, the total time spent by the user during the login phase and verification phase; and C3, the total time spent by the server during the verification phase. As shown in Table 2, Tseng et al.'s scheme is more efficient in terms of computational cost. However, Tseng et al.'s scheme is vulnerable to stolen smart card and offline

Table 2. Computational cost comparison of our scheme with other schemes.

	Proposed scheme	Liao et al.'scheme [27]	Tseng et al.'scheme [26]
C1	$3TG_{mul}+TG_H+2T_h$	$3TG_{mul}+TG_H+T_h$	$2TG_{mul}+TG_H+T_h$
C2	$8TG_{mul}+TG_H+TG_{add}+5T_h$	$5TG_{mul}+TG_H+TG_{add}+5T_h$	$3TG_{mul}+2T_h$
C3	$2TG_e+4TG_{mul}+TG_{add}+2T_h$	$2TG_e+5TG_{mul}+TG_{add}+2T_h$	$2TG_e+TG_{mul}+TG_H+TG_{add}+T_h$

https://doi.org/10.1371/journal.pone.0202657.t002

Table 3. Functionality comparisons among related multi-server authentication protocols.

	Proposed scheme	Liao et al. [27]	Tseng et al. [26]	Li et al. [20]	Lee et al. [18]	Shao et al. [17]	Lee et al. [19]
Resist stolen smart card and offline dictionary attacks	Yes	No	No	No	No	No	No
Resist replay attacks	Yes	Yes	Yes	No	No	No	No
Resist impersonation attacks	Yes	Yes	Yes	No	No	No	No
Resist server spoofing attacks	Yes	Yes	No	No	No	No	No
Resist insider attacks	Yes	Yes	No	Yes	Yes	No	Yes
Resist denial of service attacks	Yes	No	Yes	Yes	Yes	Yes	No
Perfect forwarding secrecy	Yes	Yes	No	Yes	Yes	No	No
Ensure user's anonymity	Yes	No	No	Yes	Yes	No	Yes
No verification table	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Local password verification	Yes	No	Yes	Yes	Yes	Yes	No
Proper mutual authentication	Yes	Yes	No	Yes	No	Yes	Yes

https://doi.org/10.1371/journal.pone.0202657.t003

dictionary attacks, server spoofing attacks and insider attacks and cannot provide perfect forwarding secrecy, user anonymity, proper mutual authentication and session key agreement. In our proposed scheme, the total computational cost for the user (C2) is $9TG_{mul}+TG_H+TG_{add}+5T_h$. However, similar to Liao et al.'s scheme, the user U_i can pre-compute $R_i = r_i \cdot P$ in the client, and then, the computational cost of the user (C2) requires $8TG_{mul}+TG_H+TG_{add}+5T_h$ on-line computations. It can be found that our proposed scheme has a slightly higher computational cost than Liao et al.'s scheme in C2, and the others are almost equal. However, Liao et al.'s scheme is vulnerable to stolen smart card and offline dictionary attacks and denial of service attacks and cannot provide user anonymity and local password verification.

Table 3 lists the functionality comparisons among our proposed scheme and other related schemes. It is obvious that our scheme has many excellent features and is more secure than other related schemes.

Conclusion

In this paper, we note that Li et al.'s scheme is vulnerable to stolen smart card and offline dictionary attacks, replay attacks, impersonation attacks and server spoofing attacks. Furthermore, by analyzing some other similar schemes, we find that certain types of dynamic ID-based and non-RC-dependent multi-server authentication schemes in which only hash functions are used face difficulties in providing perfectly efficient and secure authentication. To compensate for these shortcomings, we propose a novel dynamic ID-based and non-RC-dependent remote user authentication scheme for multi-server environments using pairing and self-certified public keys. The security and performance analyses show that the proposed scheme is secure against various attacks and has many excellent features. In the future, the

use of authentication for high-tech industries, such as cloud computing [42–44] and big data [44–46], will be an important area and research task.

Author Contributions

Conceptualization: Shudong Li.

Data curation: Xiaobo Wu.

Formal analysis: Shudong Li.

Funding acquisition: Shudong Li.

Investigation: Shudong Li.

Methodology: Dawei Zhao.

Project administration: Dawei Zhao.

Resources: Zhihong Tian.

Software: Aiping Li, Zhihong Tian.

Supervision: Aiping Li.

Writing – review & editing: Xiaodong Yang.

References

1. Hwang T, Chen Y, Laih CS. Non-interactive password authentication without password tables. IEEE Region 10 Conference on Computer and Communication System, 1990;1:429-431.
2. Sun HM. An efficient remote user authentication scheme using smart cards. IEEE Trans. Consum. Electron. 2000; 46(4):958–961.
3. Hwang MS, Lee CC, Tang YL. A simple remote user authentication scheme. Math. Comput. Model. 2002; 36(1-2):103–107. https://doi.org/10.1016/S0895-7177(02)00106-1
4. Das ML, Saxena A, Gulati VP. A dynamic ID-based remote user authentication scheme. IEEE Trans. Consum. Electron. 2004; 50(2):629–631. https://doi.org/10.1109/TCE.2004.1309441
5. Fan CI, Chan YC, Zhang ZK. Robust remote authentication scheme with smart cards. Computers & Security. 2005; 24(8):619–628. https://doi.org/10.1016/j.cose.2005.03.006
6. Lee SW, Kim HS, Yoo KY. Efficient nonce-based remote user authentication scheme using smart cards. Applied Mathematics and Computation. 2005; 167(1):355–361. https://doi.org/10.1016/j.amc.2004.06.111
7. Li CT, Hwang MS. An efficient biometrics-based remote user authentication scheme using smart cards. Journal of Network and Computer Applications. 2010; 33(1):1–5. https://doi.org/10.1016/j.jnca.2009.08.001
8. He D, Chen J, Hu J. An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security. Information Fusion. 2012; 13(3):223–230. https://doi.org/10.1016/j.inffus.2011.01.001
9. Li X, Niu JW, Ma J, Wang WD, Liu CL. Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. Journal of Network and Computer Applications. 2011; 34(1):73–79. https://doi.org/10.1016/j.jnca.2010.09.003
10. Lee WB, Chang CC. User identification and key distribution maintaining anonymity for distributed computer network. Journal of Computer and System Sciences. 2000; 5(4):211–214.
11. Juang WS. Efficient multi-server password authenticated key agreement using smart cards. IEEE Transactions on Consumer Electronics. 2004; 50(1):251–255. https://doi.org/10.1109/TCE.2004.1277870
12. Hsiang HC, Shih WK. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standard & Interfaces. 2009; 31(6):1118–1123. https://doi.org/10.1016/j.csi.2008.11.002

13. Sood SK, Sarje AK, Singh K. A secure dynamic identity based authentication protocol for multi-server architecture. Journal of Network and Computer Applications. 2011; 34(2):609–18. https://doi.org/10.1016/j.jnca.2010.11.011
14. Li X, Xiong YP, Ma J, Wang WD. An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. Journal of Network and Computer Applications. 2012; 35(2):763–769. https://doi.org/10.1016/j.jnca.2011.11.009
15. Liao YP, Wang SS. A secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces. 2009; 31(1):24–29. https://doi.org/10.1016/j.csi.2007.10.007
16. Hsiang HC, Shih WK. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standard & Interfaces. 2009; 31(6):1118–1123. https://doi.org/10.1016/j.csi.2008.11.002
17. Shao M, Chin Y. A novel approach to dynamic id-based remote user authentication scheme for multi-server environment. In: 2010 4th International Conference on Network and System Security (NSS 2010). IEEE Press, 2010;548–553.
18. Lee CC, Lin TH, Chang RX. A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards. Expert Systems with Applications. 2011; 38(11):13863–13870.
19. Lee CC, Lai YM, Li CT. An Improved Secure Dynamic ID Based Remote User Authentication Scheme for Multi-Server Environment. International Journal of Security and Its Applications. 2012; 6(2):203–209.
20. Li X, Ma J, Wang WD, Xiong YP, Junsong Zhang. A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments. Mathematical and Computer Modelling. 2013; 5(1-2):85–95.
21. Lee WB, Chang CC. User identification and key distribution maintaining anonymity for distributed computer network. Comput. Syst. Sci. 2000; 15(4):211–214.
22. Tsuar WJ, Wu CC, Lee WB. A flexible user authentication for multiserver internet services. Networking-JCN2001LNCS. 2001; 2093:174–183.
23. Lin C, Hwang MS, Li LH. A new remote user authentication scheme for multiserver architecture. Future Generation Computer Systems. 2003; 1(19):13–22. https://doi.org/10.1016/S0167-739X(02)00093-6
24. Geng J, Zhang L. A dynamic ID-based user authentication and key agreement scheme for multi-server using bilinear pairings. in: Proceedings of the 2008 Workshop on Power Electronics and Intelligent Transportation System. 2008;33–37.
25. Chung YH, Tseng YM. Security weakness of two dynamic ID-based user authentication and key agreement schemes for multi-server environment. in: 2009 National Computer Symposium. 2009;250–257.
26. Tseng YM, Wu TY, Wu JD. A pairing-based user authentication scheme for wireless clients with smart card. Informatics. 2008; 19(2):285–302.
27. Liao YP, Hsiao CM. A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients, Future Generation Computer Systems. 2013; 29:886–900. https://doi.org/10.1016/j.future.2012.03.017
28. Chou JS, Chen YL, Huang CH, Huang YS. Comments on four multi-server authentication protocols using smart card. IACR Cryptology. ePrint Archive 2012; 406.
29. Chuang YH, Tseng YM. Towards generalized ID-based user authentication for mobile multi-server environment, International Journal of Communication Systems. 2012; 25(4):447–460. https://doi.org/10.1002/dac.1268
30. Yeh KH, Lo NW, Li YJ. Cryptanalysis of Hsiang-Shih's authentication scheme for multi-server architecture. International Journal of Communication Systems. 2011; 24(7):829–836. https://doi.org/10.1002/dac.1184
31. Kumar A, Om H. An improved and secure multiserver authentication scheme based on biometrics and smartcard. Digital Communications and Networks. 2018; 4(1):27–38. https://doi.org/10.1016/j.dcan.2017.09.004
32. Wang CY, Xu GA, Li WT. A Secure and Anonymous Two-Factor Authentication Protocol in Multiserver Environment, Security and Communication Networks. 2018; 2018:9062675.
33. Shen H, Gao CZ, He DB, Wu LB. New biometrics–based authentication scheme for multi-server environment in critical systems. Journal of Ambient Intelligence and Humanized Computing. 2015; 6(6):825–834. https://doi.org/10.1007/s12652-015-0305-8
34. Wang CQ, Zhang X, Zheng ZM. Cryptanalysis and Improvement of a Biometric-Based Multi-Server Authentication and Key Agreement Scheme. PLoS One. 2016; 11(2):e0149173. https://doi.org/10.1371/journal.pone.0149173 PMID: 26866606

35. Reddy AG, Das AK, Odelu V, Yoo KY. An Enhanced Biometric Based Authentication with Key–Agreement Protocol for Multi-Server Architecture Based on Elliptic Curve Cryptography. PLoS ONE. 2016; 11(5):e0154308. https://doi.org/10.1371/journal.pone.0154308 PMID: 27163786

36. Chaudhry SA, Naqvi H, Mahmood K, Ahmad HF, Khan MK. An Improved Remote User Authentication Scheme Using Elliptic Curve Cryptography. Wireless Personal Communications. 2016; 90(321):1–19.

37. Yang XD, An FY, Yang P, Liu TT, Wang CF. Cross-domain Identity Authentication Scheme in Cloud Based on Certificateless Signature. Computer Engineering. 2017; 43(11):128–133.

38. Girault M. Self-certified public keys. Advances in Cryptology, Eurocrypt'91. Springer-Verlag, 1991;491–497.

39. Petersen H, Horster P. Self-certified keys concepts and applications, in: Proceedings of the 3rd Conference of Communications and Multimedia Security. Athens, 1997 September; 22–23.

40. Yu Y, Wang HM, Yin G, Wang T. Reviewer recommendation for pull-requests in GitHub: What can we learn from code review and bug assignment?. Information and Sotware Technology. 2016; 74: 204–218. https://doi.org/10.1016/j.infsof.2016.01.004

41. Luo CC, Osborne M, Wang T. An effective approach to tweets opinion retrieval. World Wide Web. 2015; 18(3):545–566. https://doi.org/10.1007/s11280-013-0268-7

42. Li T, Li J, Liu ZL, Li P, Jia CF. Differentially Private Naive Bayes Learning over Multiple Data Sources. Information Sciences. 2018; 444:89–104. https://doi.org/10.1016/j.ins.2018.02.056

43. Gao CZ, Cheng Q, He P, Susilo W, Li J. Privacy-Preserving Naive Bayes Classifiers Secure against the Substitution-then-Comparison Attack. Information Sciences. 2018; 444:72–88. https://doi.org/10.1016/j.ins.2018.02.058

44. Li J, Liu ZL, Chen XF, Tan X, Wong DS. L-EncDB: A Lightweight Framework for Privacy–Preserving Data Queries in Cloud Computing. Knowledge-based Systems. 2015; 79:18–26. https://doi.org/10.1016/j.knosys.2014.04.010

45. Li J, Chen XF, Chow SSM, Huang Q, Wong DS, Liu ZL. Multi-authority fine-grained access control with accountability and its application in cloud. Journal of Network and Computer Applications. 2018; 112:89–96. https://doi.org/10.1016/j.jnca.2018.03.006

46. Huang ZG, Liu SL, Mao XP, Chen KF, Li J. Insight of the Protection for Data Security under Selective Opening Attacks. Information Sciences. 2017; 412-413:223–241. https://doi.org/10.1016/j.ins.2017.05.031