Network and Communication Technologies; Vol. 3, No. 1; 2018
ISSN 1927-064X E-ISSN 1927-0658
Published by Canadian Center of Science and Education

A Robust Authentication Scheme for Client-Server Architecture with Provable Security Analysis

Saeed Ullah Jan[1] & Fawad Qayum[1]

[1] Department of Computer Science & IT, University of Malakand, Khyber Pakhtunkhwa, Pakistan

Correspondence: Saeed Ullah Jan, Department of Computer Science & IT, University of Malakand, Khyber Pakhtunkhwa, Pakistan. E-mail: saeedullah@uom.edu.pk

Received: April 8, 2018 Accepted: April 18, 2018 Online Published: April 27, 2018

doi:10.5539/nct.v3n1p6 URL: https://doi.org/10.5539/nct.v3n1p6

Abstract

Client-server computing is the analytical development of compatible programming with significant supposition and the detachment of a massive program into its fundamental parts ("modules"), which can create the chance for extra enhancement, inconsiderable improvement, and prominent maintainability. In client-server computing, total extensive modules don't need to be accomplished within the similar memory space totally but can execute independently on a suitable hardware and software platform according to their behavior. The user authentication is the dominant constraint for client-server computing that limits the illegitimate right of entry into the main workstation. This research is mainly focused on the design of a robust authentication scheme for client-server architecture computing. It carries some additional features like security, virtualization, user's programs security, individuality supervision, integrity, control access to server and authentication. The proposed background also delivers the characteristic supervision, mutual authentication, and establishment of secure session key among users and the remote server.

Keywords: discrete logarithm problem, BAN-Logic, wired communication media, scheduling, access control, identity management

1. Introduction

The client-server architecture allows a group of shared computing resources' access to the community, anytime and at any place via the internet. These resources are available via these models such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The demand of this client-server architecture is currently in its peak due to its integrated qualities as a demonstration of huge processing power, quick management of a large amount of data and approachability of exclusive hardware resources in much-reduced cost. Further, relaxed availability, well performance and the capability to measure each and every matter are supreme in this technological era (Lamport, 1981; Liao & Wang, 2009; Hsiang & Shih, 2009). But in this environment, there are some thoughtful anxieties about it that must be faced at the beginning to make it more reliable, stable and manageable. For this purpose, security is one of the non-negligible things of client-server architecture. The security matter exists at both peers, means cloud service providers and cloud service users. Both have confrontations to protect themselves from possible security risks for securing sessions, affairs or computations. The server is answerable to deliver a secure and well-protected framework to its clients concerning the protection of sensitive information, authorizations, and applications. Whereas, servers are also authoritative for using difficult pin codes, and robust authentication schemes for the appropriateness of secure and continuous services (Lamport, 1981; Liao & Wang, 2009; Hsiang & Shih, 2009).

A client-server computing commonly regulates the task scheduling fragment of the application program; authenticate data recorded by a user, post a request to the remote server platforms (Jan, 2017). The user interfaces part of the application program design for end users to understand and interface with. Moreover, the Client side also accomplishes the bounded devices that the end user relates with such as the monitor, mouse, keyboard, CD-ROM, workstation, printer, scanner, CPU, Floppy and other peripherals (Wu, Xu, Kumari, Xiong, & Abdulhameed, 2015). Afterward, the server gets demands from clients; accomplishes record repossession, renews and regulates data integrity and posts replies to client demands.

The server works as software powerhouse that accomplishes common means such as software, databases, printing devices, communication line, or high powered CPU. The key aim of the server is to complete the backend

responsibilities that are mutual to related applications. Some software allows applications to communicate independently for processes or programs with each other. Network Operating System facilitates service area, such as direction-finding, delivery, messaging, communication supervision service, a guideline for different tasks (D. Wang, & P. Wang, 2014). Subsequently, the somatic linking has been recognized and Transfer Control Protocol (TCP) is carefully chosen. Distributed computing protocol is mandatory before the client can take advantages of the network facilities. A distributed computing environment protocol requires a strong authentication protocol to request for a routine and the server responses securely (D. Wang, & P. Wang, 2014). So that firstly it might provide a straightforward application usage, secondly the applications must not be in isolation, and not be a monolithic system, and lastly, the applications programs must not be complicated and the supporting technology must not base on a centralized control model.

With the rapid expansion of online information accessing and client-server network usage, the requirement of securing sensitive personal information of a user either locally or executing a server remotely turn out to be ever more necessary. Therefore, a concept called discrete logarithm problem (Eric Bach, 1984) is a powerful solution which combines some benefits from conventional cryptography and other from basic security. Here, we demonstrate an efficient and strong authentication scheme which can mitigate the concept of discrete logarithm problem for security and the newly designed protocol shall be worked for the authentication of a legal user in the client-server environment, with the assumption that the server is more secure. The discrete logarithm problem characteristics are not only used for user authentication but can also for cryptographic key generation technique. The concept of Discrete Logarithmic Problem (Eric Bach, 1984) is explained as under:

The discrete logarithm problem characteristics are not only used for user authentication but can also for cryptographic key generation technique. Discrete logarithmic problem is very easy to compute $h=g^x$ for a given x, but very hard to find x given h and g.

Group G is a set with operations and each element has an inverse. Suppose G represents a group multiplicatively and g for cyclic sub-group i.e. g ε G. Then Discrete Logarithmic Problem for G is written as Assumed g ε G and a ε (g), the integer x is g^x= an (Eric Bach, 1984). The x is the discrete logarithm of a base g and is represented by $x = ind_g a$. Suppose p denotes a big prime while g be GF(p) "generator of the multiplicative group". Then the function f can be defined as: f: {1, ……, p-1}→$GF(p)^*$ by $f(x) = g^x \pmod p$. This function is easy to compute by using the binary expansion of x. Let $x = \sum_{i=0}^{k} \epsilon_i 2^i$ with $\epsilon_i = 0$ or 1. Then

$g^x = \prod_{\epsilon_i=1} g^{2^i} \pmod p$

By squaring and multiplication repeatedly, one can easily compute the right-hand side (RHS) using at most 2k multiplicative modulo p. But on the other method, changing the function f, requiring an algorithm for the DLP in $GF(p)^*$, and thus widely used to be intractable for large p.

In other words the said function can also be expressed as: Let a simple element say g from multiplicative-group fields GF(q), then the Discrete Logarithmic Problem (Odlyzko & Andrew, 1984) for a non-zero component will be u**ϵ**GF(q) represent that the integer k in the finite fields of values "$1 \le k \le q-1$", so that $u = g^k$.

The famous function of computing is the discrete logarithmic function (Odlyzko & Andrew, 1984) in the finite fields of a set that has shown extra capabilities for the last few decades due to its suitability in cryptography. Many cryptographic functions used by mathematician and computer scientists are vulnerable to several threats, but when the discrete logarithm problem was revealed the information sharing has more secure. It seems that in order to secure information from all known attacks using this method, the element of n in multiplicative-group fields $GF(2^n)$ is adopted in cryptosystem which can easy to calculate but very hard to find the element chosen from n. Similarly, the large values of Discrete Logarithmic Problem (Odlyzko & Andrew, 1984) in multiplicative-group fields $GF(2^n)$ is considerably easy to calculate but appeared to offer comparatively extraordinary levels of security. In many services provider computer systems, the user's passwords or PIN codes are stored in a specified file, which appears not secure and anyone who becomes access to the password's table is capable to freely and easily impersonate any genuine user. So the password's table needs much attention to be protected from an unauthorized user. Therefore, this methodology is effective and efficient for securing it, because the multiplicative-group fields GF(q) and a primitive values g ε GF(q) are selected, where x is an integer of many high values, put in $f(x)=g^x$ and made password's table or file public, no one can guess or find it at any stage. Also, anyone attempt for accessing password's file on the computer and pretending to ith user would have to calculate q_i by expressing only the value of g; i.e., he/she has to explain the Discrete Logarithmic Problem (Odlyzko & Andrew, 1984) in the group fields GF(q) which is not possible for them to compute the value.

2. Related Work

Since the first authentication scheme was designed by Lamport in 1981 (Lamport, 1981) using a simple PIN code or a simple password for remote user authentication; later on, the community has focused considerable attention to this important research area. So for Wang and Liao (Liao & Wang, 2009) have presented ID-based authentication scheme by means of lightweight cryptographic functions, i.e. bit-wise X-OR operation and a single-way digital hash function to deliver mutual authentication and session key arrangement. In addition, the Wang and Liao (Liao & Wang, 2009) protocol is based on 2-factor and the idea of the nonce. The Wang and Liao (Liao & Wang, 2009) claimed that their protocol assures computation effectiveness and individual anonymity. Then Hsiang and Shih (Hsiang & Shih, 2009) proved that Wang and Liao (Liao & Wang, 2009) protocol is defenseless in contradiction of impersonation, insider, and server spoofing attacks and might do not deliver mutual authentication. Hsiang and Shih (Hsiang & Shih, 2009) then presented a medication which is planned to restore the security weaknesses they exposed. They succeeded the similar level of computation effectiveness by applying a single-way digital function and XOR-operation in their scheme. Next, Sood *et al.'s* (Sood, Sarje, & Singh, 2011) used a two-server model design in which dissimilar points of confidence are allocated to the main services provider computer, and the client's authenticate information is spread among a couple of servers, called the services supplier server and controller server. As controller server comprises all users' confidential record and is not openly accessible to the clients, it does not have appropriately under attack. However, the diffidence of the protocols suggested in (Li, C. C.Lee, Liu, & C. W. Lee, 2011) and (Hsiang & Shih, 2009) was verified by researchers (X. Li, Qiu, Zheng, Chen, & J. Li, 2010; Li, Xiong, Ma, & Wang, 2012; Juang, Chen, & Liaw, 2008) correspondingly, which exposed that confrontation to impersonation, replay, stolen smart card and leak of verifier attacks could not be delivered.

Later, Lee and Chang (Chang & Lee, 2012) demonstrated a single-sign-on-based authentication scheme for shared networks. The idea of single-sign-on can permit legitimate users to use a unitary symbol to access distributed service providers. The client-server architecture is assumed in the Chang and Lee (Chang & Lee, 2012) scheme and heavyweight exponential computation are implemented to convey the tough security density of their protocol and the security parameters of their protocol appeared unambiguous and considered to be a robust one. However, the researchers in (Yang, Wong, Wang, & Deng, 2008) found two flaws i.e. user impersonation and credential recovering attacks which might rise well against (Chang & Lee, 2012). Another scheme was presented by Juang *et al.'s* (2012) based on Elliptic Curve Cryptography (ECC) and Symmetric Cryptographic Functions (SCF) using a smart card for remote user authentication. They claim that their protocol might gain identity protection; guarantee for the session key, confront low networking features cost and resists insider attack because of ECC and SCF. But, all these announcements cannot confirm by (Tsai, Lo, & Wu, 2013; Li et al., 2010). Later, Tsai *et al.'s* (Tsai, Lo, & Wu, 2013) originate that Li *et al.'s* (2010) protocol is defenseless toward de-synchronization attack. Furthermore, personal sensitive data about a user "update mechanism" in Li et *al.'s* (2010) protocol is not properly addressed and also no effective registration database has been developed. So, Tsai *et al.'s* (Tsai, Lo, & Wu, 2013) validated all the loopholes by designing an anonymous authentication protocol. The different characteristic of Tsai *et al.'s* (Tsai, Lo, & Wu, 2013) authentication scheme is that; it cloud doesn't need to preserve a registration record for its clients, which makes the protocol appropriate for the distributed environment.

Wang *et al.'s* in different research articles (Wang & Ma, 2012; D. Wang, Ma, P. Wang, & Chen, 2012; D. Wang, Ma, & P. Wang, 2012b; D. Wang & P. Wang, 2013; Y, Wang, 2012) present a remarkable learning to examine the trust among smart cards and terminal; that is, whenever an attacker gets a lost smart card, the chance of user's information is being compromised at any stage and at level. So, based on Common Adversary Model (CAM) containing three types of attackers and four important points are presented in the protocols (Wang & Ma, 2012; D. Wang, Ma, P. Wang, & Chen, 2012; D. Wang, Ma, & P. Wang, 2012b; D. Wang & P. Wang, 2013; Y, Wang, 2012): (a) a private key based schemes are secure against the type I and II attackers, but not against a type III attacker; (b) a public key schemes are secure against type I, II and III attackers; (c) a public key HMQV-based schemes are secure against type I and II attackers, but not against the type III attacker; and (d) a public key based PSCAV-based schemes are secure against type I, II and III attackers. Then, Wang et al. (Wang & Ma, 2012; D. Wang, Ma, P. Wang, & Chen, 2012; D. Wang, Ma, & P. Wang, 2012b; D. Wang & P. Wang, 2013; Y, Wang, 2012) found that PSCAb has many practical drawbacks, and PSCAV is defenseless in the type III attacker. Moreover, the authors examined many password-based authentication schemes and offered 12 estimation principles for it. Wang (Wang & Ma, 2012; D. Wang, Ma, P. Wang, & Chen, 2012; D. Wang, Ma, & P. Wang, 2012b; D. Wang & P. Wang, 2013; Y, Wang, 2012)also presented the confidence of two authentication protocols of Leu and Hsieh (Hsieh & Leu, 2012) and found that their scheme is defenseless to the offline dictionary. Additionally, the authors proposed a comparative study of "two-factor authentication schemes using smart cards" and "common-memory device- based two-factor schemes" under two self-defined adversary models.

Then, Huan *et al.'s* (2013) designed two detailed security mechanisms for distributed environment using Personal Identification (PIN) code for authentication using a smart card i.e. an attacker that can store a similar data in the smart card, and the other is, an attacker that can store different data in the smart card. So, two threats were identified in this regard, first, services provided to the legitimate user by an authentication scheme is difficult in their scheme, and second, the services delivered by their schemes was not consistent and also showed countermeasures problem. In another scheme, Wang *et al.'s* proposed a 5-phase authentication scheme containing registration, login, verification, password change, and card revoking phases. They examined the probability of designing an anonymous, two-factor authentication scheme with the concept of "*Madhusudhan Mittal' Evaluation Set*"(D. Wang & P. Wang, 2014). They also presented the characteristics of offline request submitter's password change facility and strong resistance to stolen or loss smart card attack which are hard to realize simultaneously. Later, in (D. Wang & P. Wang, 2013) they investigated all the weaknesses among system efficiency and user anonymity and scrutinized a significant result: a public key infrastructure (PKI) technique is essentially crucial for a two-factor authentication scheme with user anonymity. But they used cryptographic method for their schemes. Moreover, in other scheme they confirmed that the password-based scheme of different researchers like (X. Li, Qiu et al., 2010; X. Li, Xiong et al., 2012; Juang et al., 2008) cannot resist Denial-of-Service (DoS) and offline password guessing attacks and failed to deliver strong user's anonymity as well as forward secrecy and mutual authentication.

3. Review Analysis of Scheme (Hassan, Eltayieb, Elhabob, & Li, 2017)

The scheme of (Hassan, Eltayieb, Elhabob, & Li, 2017) is based on certificateless public key cryptography for the client-server environment by Hassan et al.'s in 2017. The review analyses of different phases are as under:

Table 1. Notations used in (Hassan, Eltayieb, Elhabob, & Li, 2017)

Notations use by (Hassan, Eltayieb, Elhabob, & Li, 2017) and its description			
ID_c	Participants' Identity	D_{IDc}	Client Private Key
P_{IDc}	Client Public Key	P_{IDs}	Server Public Key
ID_j	Challenged Identity	P_{pub}	Server Master Key
G_1	Cyclic Additive Group	G_2	Cyclic Multiplicative Group
p	Generator of G_1	q	Prime Order for G_1 and G_2
k	Parameter	s	Master secrete key
e	$G_1 \times G_1 \rightarrow G_2$	H_i	Hash function

3.1 Initialization Phase

The computations performed in the initialization phase of the scheme are as under:

A- Setup (1^k): This step of the scheme is performed by the key generator center (KGC), *k* called a security parameter is taken by Key-Generator-Centre (KGC) while the other parameters generated are as follows:

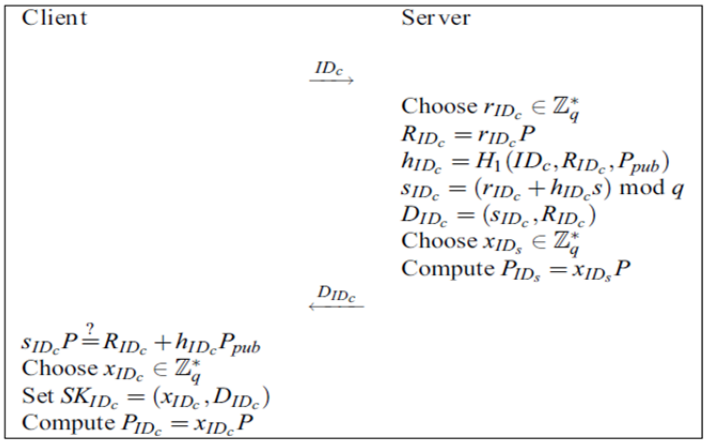

Figure 1. Initialization Phase

i. G_1 and G_2 are in the order of q and bilinear pairing e: G_1 x $G_2 \rightarrow G_2$ where p is a generator of G_1.

ii. The master secretes key $s \in Z_q^*$ and master public key $P_{pub} = sP$ is calculated.

iii. After it, the public key $P_{IDs} = x_{IDs}P$ is computed, where $x_{IDs} \in Z_q^*$ and secure hash-functions $H_1 : \{0, 1\}^* \times G_1 \times G_2 \rightarrow Z_q^*$, $H_2 : G_1 \times \{0, 1\}^* \times Z_q^* \times G_1 \times G_1 \times G_1 \rightarrow Z_q^*$, $H_3 : \{0, 1\}^* \times G_1 \times G_1 \times G_1 \times Z_q^* \times G_1 \times Z_q^* \rightarrow Z_q^*$ and $H_4 : \{0, 1\}^* \times G_1 \times G_1 \times G_1 \times Z_q^* \times G_1 \times Z_q^* \rightarrow G_1$ are selected Extracting partial, private and public keys.

3.2 Authentication Phase

Herein the authentication phase of the scheme both server and client exchange the key to identify legality of both the peers and become authenticated as shown in the figure below:-

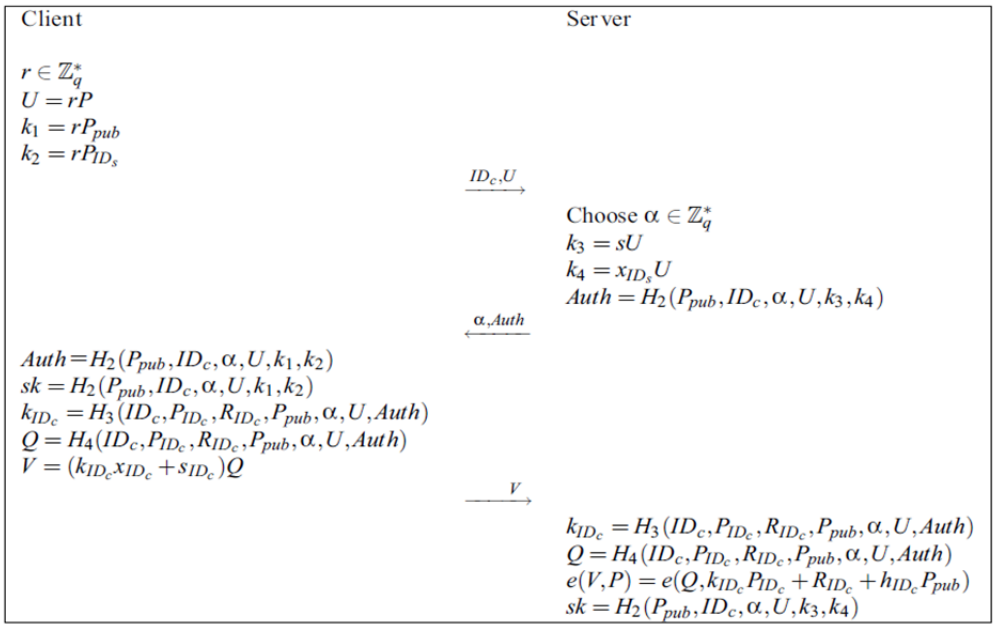

Figure 2. Authentication and key exchange phase

3.3 Cryptanalysis of the scheme (Hassan, Eltayieb, Elhabob, & Li, 2017)

The (Hassan et al., 2017) claim that their scheme is well designed and secure against all known attacks but the insight study shows that (Hassan et al., 2017) still has some security loopholes in terms of replay, Denial-of-Service (DoS) and impersonation attacks and also failed to show strong user anonymity. The identity used in the scheme is not dynamic because the attacker can guess it anywhere when user login with the same identity. Similarly, the attacker can also extract some useful information at the authentication phase of the scheme (Hassan et al., 2017), due to lack of timestamp. If an attacker failed by extracting such useful information, definitely he/she can hang the useful resources of the server by launching a denial-of-service attack. Similarly, the scheme clock synchronized in 2, 3 round strip for authenticating the server and establishing mutual authentication. Therefore, the said scheme is failed for the secure authentication of client-server peers securely, accurately and anonymously.

4. The Proposed Scheme

The proposed robust authentication scheme working both for wired and wireless communication channels and specially designed for client-server architecture which mitigates the idea of Discrete Logarithmic Problem/Function (DLP) and consists of five phases: the registration, login and authentication, password change and card revocation. The organization of the paper is planned as: in this section, basic notations, terminologies and different phases of the scheme will be discussed in detail; section 5 the proposed scheme is formally analyzed

using (Burrows, Abadi, & Needham, 1995; Mart´ın & Andrew, 1998) and (Chuang & Chen, 2014) techniques and in section 6 the informal analysis of the proposed authentication scheme will be presented using general intelligence and experience while in the last section the performance comparison of the scheme will be given comprehensively.

Table 2. Notations used for the proposed scheme

Notations or Preliminaries and its Description			
1. S, IDs	the remote server and its Identity	2. U_i, ID_u	the user and its Identity
3. p, q	Large Prime numbers	4. g	Multiplicative group
5. x	secret key	6. PW_{ib}:	Password of User Ui
7. a_i, b_i, r_i, α, β	Random Numbers	8. T	Time Stamp
9. SK_u, SK_s	session keys	10. h(.), h1(.)	hash-functions
11. l	parameter	12: A	An Adversary
13. →	an insecure path	14: a? = b	whether an equal to b

4.1 Registration Phase

The main computer "server" selects two high-scale prime numbers p and q and p = 2q + 1, g a multiplicative group, G in the direction of n, a controlling secret key x and h(.), $h_1(.) : \{0, 1\}^* \rightarrow \{0, 1\}^l$. The following steps are performed in this phase:

(1): $U_i \Rightarrow S$: ID_u, HPW_{ib}

The user U_i chooses identity ID_u, password PW_{ib} and random integer b_i, calculate $HPW_{ib} = h(PW_{ib}||b_i)$, and directs ID_u and HPW_{ib} to the server over a private channel (⇒).

(2): $S \Rightarrow U_i$: smart card

The server checks ID_u is available in the record. If identity ID_u is not present in the record of the server, the server then selects a big integer N_i and random integer a_i and computes:

$Z_1 = h(x||a_i) \oplus h(ID_u||HPW_{ib})$, $Z_2 = h(ID_u||x||N_i) \oplus HPW_{ib}$ and stored the values $\{ID_u, N_i\}$ in the record table of the server; while the server stores Z_1, Z_2, g, p, h (.) in the storage portion of a smart card and issued to user via a secure path, as shown below:

(3): $U_i \Rightarrow$ card: Z_3

In this step of the registration, the legitimate user computes:

$Z_3 = h(ID_u||PW_{ib}) \oplus b_i$ and also stores Z_3 in the smart card.

4.2 Login Phase

(1): U_i provides smart-card into a machine and gives ID_u with PW_{ib} then calculates:

$b_i^{/} = h(ID_u||PW_{ib}) \oplus Z_3$.

(2): The card generates two high entropy random integers r_i and $\alpha \in [1, q-1]$ and calculates $HPW_{ib} = h(PW_{ib}||b_i^{/})$, $R_i = Z_1 \oplus r_i$, $Q_1 = h(ID_u||HPW_{ib}) \oplus r_i$, $Q_2 = g^{\alpha} \% P$, $k_i = Z_2 \oplus HPW_{ib}$ and $Q_3 = h(ID_u||Q_2||k_i||r_i||a_i)$

(3): $S \rightarrow U_i$: $Message_1 = \{R_i, a_i, E_{Q1}(ID_u||Q_2||Q_3||r_i)\}$

The user U_i encrypts ID_u, Q_2, Q_3, and r_i with Q_1 and sends it with R_i and a_i to the server S via an insecure channel.

Client Side Computation	Server Side Computation
Chooses 2 prime numbers p and q where p=2q+1 from a group g called a multiplicative group of direction n	
Selects b_i a high entropy random integer	
Selects ID_u and PW_{ib},	
Computes: $HPW_{ib}=h(PW_{ib}\|\|b_i)$	
$S = (HPW_{ib} \oplus ID_u)$	
$\{ID_u, HPW_{ib}\}$ →	Chooses a big integer N_i, random integer a_i and a secrete key x $Z_1=h(x\|\|a_i)\oplus(ID_u\|\|HPW_{ib})$ $Z_2= h(ID_u\|\|x\|\|N_i)\oplus HPW_{ib}$
← $\{Z_1, Z_2\}$	
$Z_3= h(ID_u\|\|PW_{ib})\oplus b_i$ Finally the memory of smart card contains {S, Z_1, Z_2, Z_3, g, p, a_i and h(.)}	

Figure 3. Computation of Registration Phase

4.3 Authentication Phase

The following steps are involved during authentication phase:

(1): After receiving $Message_1$ from user, the server S calculates $Q_1'=h(x\|\|a_i)\oplus R_i$, decrypts $E_{Q1}(ID_u\|\|Q_2\|\|Q_3\|\|r_i)$ with Q_1', and get ID_u', Q_2', Q_3' and r_i'. The server checks the memory for ID_u' and the nonce N_i. If the identity doesn't confirm, the server terminate the session and if so the server computes:

Client Side Computation	Server Side Computation
Login:	
Inputs ID_u with PW_{ib}	
Computes $b_i =h(ID_u\|\|PW_{ib})\oplus Z_3$	
Generates r_i and α and computes:	
$HPW_{ib}=h(PW_{ib}\|\|b_i')$	
$R_i = Z_1\oplus r_i$,	
$Q_1 = h(ID_u\|\|HPW_{ib})\oplus r_i$	
$Q_2 = g^{\alpha} \% P$	
$k_i = Z_2 \oplus HPW_{ib}$	
$Q_3 = h(ID_u\|\|Q_2\|\|k_i\|\|r_i\|\|a_i\|\|T_1)$	
$Message_1= \{R_i, a_i, E_{Q1}(ID_u\|\|Q_2\|\|Q_3\|\|r_i), T_1\}$ →	
Authentication:	$T'-T_1 \le \Delta T$ Computes $Q_1'=h(x\|\|a_i)\oplus R_i$ Decrypts $E_{Q1}(ID_u\|\|Q_2\|\|Q_3\|\|r_i)$ with Q_1', and get ID_u', Q_2', Q_3' and r_i' Checks the memory for ID_u' and the nonce N_i Computes $k_i'=h(ID_u'\|\|x\|\|N_i)$ Checks $Q_3'?=h(ID_u'\|\|Q_2'\|\|k_i'\|\|r_i'\|\|a_i)$ Generate β and a_i', computes: $Q_4=g^{\beta} \bmod p$, $Q_5=(Q_2')^{\beta} \bmod p$ $Q_6=k_i\oplus h(r_i')$, $Q_7= h(x\|\|a_i')$ $sk_s=h_1(ID_u'\|\|Q_1'\|\|Q_2'\|\|Q_4\|\|Q_5\|\|k_i'\|\|T_2)$ $Q_8=h_1(ID_u'\|\|Q_1'\|\|Q_2'\|\|Q_4\|\|Q_5\|\|Q_6\|\|Q_7\|\|sk_s\|\|a_i')$
← $Message_2= \{E_{Q6}(Q_4\|\|Q_7\|\|Q_8\|\|a_i'), T_2\}$	
$T_1'-T_2 \le \Delta T$	
Computes $Q_6'=k_i\oplus h(r_i)$	
Decrypt the message and get Q_4', Q_7', Q_8' and a_i''	
Computes: $Q_5'=(Q_4')^{\alpha} \bmod p$	
$sk_u=h_1(ID_s\|\|Q_1\|\|Q_2\|\|Q_4'\|\|Q_5'\|\|k_i)$	
$Q_8''=h_1(ID_s\|\|Q_1\|\|Q_2\|\|Q_4'\|\|Q_5'\|\|Q_6'\|\|Q_7'\|\|sk_u\|\|a_i'')$	
Checks $Q_8'?=Q_8''$	
Computes: $Z_1'=h(ID_u\|\|h(PW_{ib}\|\|b_i'))\oplus Q_7'$	
Replace Z_1 and a_i with Z_1' and a_i''	

Figure 4. Computations of Login and Authentication Phase

$k_i^{/}=h(ID_u^{/}||x||N_i)$ and checks

$Q_3^{/}?=h(ID_u^{/}||Q_2^{/}||k_i^{/}||r_i^{/}||a_i)$, if not matched the server terminate the session.

(2): The server next generates two arbitrary numbers $\beta \in [1, n-1]$ and $a_i^{/}$, computes:

$Q_4=g^{\beta} \bmod p$, $Q_5=(Q_2^{/})^{\beta} \bmod p$,

$Q_6=k_i \oplus h(r_i^{/})$, $Q_7= h(x||a_i^{/})$,

$sk_s=h_1(ID_u^{/}||Q_1^{/}||Q_2^{/}||Q_4||Q_5||k_i^{/}||T_2)$ and

$Q_8= h_1(ID_i^{/}||Q_1^{/}||Q_2^{/}||Q_4||Q_5||Q_6||Q_7||sk_s||a_i^{/})$.

(3): $S \rightarrow U_i$: $Message_2=\{E_{Q6}(Q_4||Q_7||Q_8||a_i^{/})\}$

(4): The user after this receiving M_2, decrypt it and computes

$Q_6^{/}=k_i \oplus h(r_i)$ and get $Q_4^{/}$, $Q_7^{/}$, $Q_8^{/}$ and $a_i^{//}$. The user computes:

$Q_5^{/}=(Q_4^{/})^{\alpha} \bmod p$,

$sk_u=h_1(ID_s||Q_1||Q_2||Q_4^{/}||Q_5^{/}||k_i)$,

$Q_8^{//}=h(ID_s||Q_1||Q_2||Q_4^{/}||Q_5^{/}||Q_6^{/}||Q_7^{/}||sk_u||a_i^{//})$ and checks $Q_8^{/}?=Q_8^{//}$.

If not matched, the user terminates the session, otherwise, user U_i uses sk_u as the session key.

(5): The user after this calculates

$Z_1^{/}=h(ID_u||h(PW_{ib}||b_i^{/})) \oplus Q_7^{/}$ and interchanges Z_1 and a_i with $Z_1^{/}$ and $a_i^{//}$.

The server decrypts Q_4, Q_7, Q_8 and $a_i^{/}$ using Q_6 and sends $Message_2$ to the user.

This time the server use sk_s as the session secret key.

4.4 Password Change Phase

When the U_i desires to change his or her password, provide ID_u and PW_{ib}, the following steps are performed:

(1): After sending the message $Message_1$ to the server, a change of password demand also sends. First U_i become authenticated and then relays $Q_9= h(ID_u||Q^{\square}_1||Q^{\square}_2||k^{\square}_i||r^{\square}_i||a_i)$ and request for permission.

(2): If $Q_9 = h(ID_u||Q_1||Q_2||k_i||r_i||a_i)$ passed by the user, then enter a new password message is displayed PW_{ib}^{new}. At this stage the smart card chooses a random number b_i^{new} and computes:

$Z_1^{new} = Z_1 \oplus h(ID_u||h(PW_{ib}||b^{\square}_i)) \oplus h(ID_u||(PW_{ib}^{new}||b_i^{new}))$

$Z_2^{new} = Z_2 \oplus h(PW_{ib}||b^{\square}_i) \oplus h(PW_{ib}^{new}||b_i^{new})$, $Z_3^{new} = h(ID_u||PW_{ib}^{new}) \oplus b_i^{new}$

(3): The values of Z_1, Z_2, and Z_3 replaced by Z_1^{new}, Z_2^{new}, and Z_3^{new}. For more detail, about change of password phase, visit Jan S.U, 2017)

4.5 Card Revocation Phase

A legitimate user U_i if loss his/her smart card, can easily demand another by means of some credentials like $N_i^{new} = N_i+1$ and stored $\{ID_u, N_i^{new}\}$ in the database of smart card and the owner might issue a new smart card to the user and follow the registration phase.

5. Formal Security Analysis

5.1 BAN Logic

In the first part of this section, formal verification of the scheme is performed using BAN logic of authentication (Burrows, Abadi, & Needham, 1995). BAN logic has many stages in the form of different mathematical formulas that are mandatory to logically prove any protocol (authentication scheme). These steps are given in the following Table 3:

Table 3. BAN Logic of Authentication Protocol Steps

PROTOCOL STEPS	DESCRIPTION
$P \rightarrow Q: message$	The peer P sends a *message* to peer Q
A⟶B: {A, Kab} Kbs	Peer B knows a key K_{bs} and sends key K_{ab} to peer A.
$A \rightarrow B: \{A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{bs}}$	Peer A and peer B identify key K_{bs} and key K_{ab} is transmitted among both peers.
$B \lhd \{A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{bs}}$	Peer B *sees* the links among peer A and B through key K_{ab} and key K_{bs} is sent among both peers
$A \mid \equiv A \overset{K}{\leftrightarrow} B,\ B \mid \equiv A \overset{K}{\leftrightarrow} B$	Peer A *believes* itself and exchanging things or materials using key K and vice versa
$A \mid \equiv B \mid \equiv A \overset{K}{\leftrightarrow} B,\ B \mid \equiv A \mid \equiv A \overset{K}{\leftrightarrow} B$	Peer A *believes* peer B *believes* that peer A exchange things and materials peer B through key K and the same for the second case.
$A \mid \equiv \overset{K}{\rightarrow} B$	Peer A *believes* the exchange of message using key K to peer B
$A \mid \equiv A \overset{N_a}{\rightleftharpoons} B$	Peer A believes the exchange of None (some secrets) among peer A and B

Goal1: $U \mid\equiv S \overset{sku}{\longleftrightarrow} U$

Goal2: $S \mid\equiv U \mid\equiv S \overset{sks}{\longleftrightarrow} U$

Goal3: $U \mid\equiv S \overset{sks}{\longleftrightarrow} U$

Goal4: $U \mid\equiv S \mid\equiv S \overset{sku}{\longleftrightarrow} U$

The idealization of the Scheme

Message 1: $U \rightarrow S: \ R_i, a_i, E_{Q1}(ID_i||Q_2||Q_3||r_i||T_1): \ \{R_i, a_i, E_{C1}(ID_i||Q_2||Q_3||r_i||T_1)\}_x$

Message 2: $S \rightarrow U: E_{Q6}(Q_4||Q_7||Q_8||a_i^{/}), T_2: \ \{E_{Q6}(Q_4||Q_7||Q_8||a_i^{/}), T_2\}_x$

Assumptions of the Scheme

Assumption 1: $U \ \mid\equiv \square (T_1)$

Assumption 2: $S \ \mid\equiv \square (k_i, r_i, a_i, T_1)$

Assumption 3: $U \mid\equiv S \overset{x}{\longleftrightarrow} U$

Assumption 4: $S \ \mid\equiv S \overset{x}{\longleftrightarrow} U$

Assumption 5: $U \ \mid\equiv S \overset{sks=h1(IDi/||Q1/||Q2/||Q4||Q5||ki/)}{\longleftrightarrow} U$

Assumption 6: $S \ \mid\equiv S \overset{sku=h1(IDi||Q1||Q2||Q4/||Q5/||ki)}{\longleftrightarrow} U$

Assumption 7: $U \ \mid\equiv S \ \Rightarrow \ (Q_4, k_i)$

Assumption 8: $S \ \mid\equiv U \ \Rightarrow \ (T_1)$

Next, take Message 1 and Message 2 as,

Message 1: $U \rightarrow S: \ R_i, a_i, E_{Q1}(ID_i||Q_2||Q_3||r_i||T_1): \ \{R_i, a_i, E_{Q1}(ID_i||Q_2||Q_3||r_i||T_1)\}_x$

By applying the seeing rule,

S1: $S \lhd R_i, a_i, E_{Q1}(ID_i||Q_2||Q_3||r_i||T_1): \{R_i, a_i, E_{Q1}(ID_i||Q_2||Q_3||r_i||T_1)\}_x$

According to S1, A3, and Q_1, the following result will be obtained

S2: $S \mid\equiv U\sim (R_i, a_i, E_{Q1}(ID_i||Q_2||Q_3||r_i||T_1))$

According to A1, S2, R4, and R2

S3: $S \mid\equiv U \mid\equiv (R_i, a_i, E_{Q1}(ID_i||Q_2||Q_3||r_i||T_1))$

Where T_1 is the client side time

According to A_7, S_3, and Jurisdiction rule

S4: $S \mid\equiv (R_i, a_i, E_{Q1}(ID_i||Q_2||Q_3||r_i||T_1))$

According to A_5, S_4, and session key rule

S5: $S \mid\equiv U \mid\equiv S \xleftrightarrow{sku=h1(IDi||Q1||Q2||Q4/||Q5/||ki)} U$ Achieved (Goal 2)

According to A_7, S_5, and R_4 rule

S6: $S \mid\equiv S \xleftrightarrow{sks=h1(IDi/||Q1/||Q2/||Q4||Q5||ki/)} U$ Achieved (Goal 1)

The 2nd idealized message as:

Message 2: $S \rightarrow U: E_{Q6}(Q_4||Q_7||Q_8||a_i^{/}), T_2: \{E_{Q6}(Q_4||Q_7||Q_8||a_i^{/}), T_2\}_x$

By the application of seeing the rule,

S7: $U \lhd S \rightarrow U: E_{Q6}(Q_4||Q_7||Q_8||a_i^{/}), T_2: \{E_{Q6}(Q_4||Q_7||Q_8||a_i^{/}), T_2\}_x$

According to S_7, A_4, and R_1

S8: $U \mid\equiv S \sim (Q_3, a_i \,||\, T_2)$

According to A_2, S_8, R_4, and R_3 rules, the following result be obtained

S9: $U \mid\equiv S \mid\equiv (Q_3, a_i \,||\, T_2)$

T_2 is server timestamp, so

According to A_6, S_9, and R_4 rule

S10: $U \mid\equiv (Q_3, a_i \,||\, T_2)$

According to A_4, S_{10}, and session key rule

S11: $U \mid\equiv S \mid\equiv S \xleftrightarrow{sku=h1(IDi||Q1||Q2||Q4/||Q5/||ki)} User$ Achieved (Goal 4)

According to A8, S11, and Jurisdiction rule

S12: $U \mid\equiv S \xleftrightarrow{sks=h1(IDi/||Q1/||Q2/||Q4||Q5||ki/)} U$ Achieved (Goal 3)

5.2 GNY Logic

Gong-Needham-Yahalom (Chuang & Chen, 2014) Logic is another way of formally proving a cryptographic protocol. It is just like that of BAN logic. In this study, (Chuang & Chen, 2014) is used to formally evaluate the aforesaid security authentication scheme. The main features of (Chuang & Chen, 2014) for the authentication scheme are as under:

Therefore, to fit the GNY logic (Odlyzko, A. M, 1984) for this authentication scheme, the transformation of different formulas will be shown as given below:

1) Client → Server: $\{R_i, a_i, (ID_u||Q_2||Q_3||r_i), T_1\}_{Q1}$

2) Server → Client: $\{Q_4, Q_7, h_1(ID_u^{/}||Q_1^{/}||Q_2^{/}||Q_4||Q_5||Q_6||Q_7||sk_s||a_i^{/})_{Q6}, a_i^{/}, T_2\}$

3) Server → Client: $\{h_1(ID_u^{/}||Q_1^{/}||Q_2^{/}||Q_4||Q_5||Q_6||Q_7||sk_s||a_i^{/})\}$

In the next step, the goals will be achieved. For example,

Goal – 1: Server believes the message in the first round is recognizable.

Server$\mid \equiv\phi (\{ID\}_{Q1}, \{ R_i, a_i,||ID_u||(g^{\alpha} \% P)||r_i\}_{q1})$

Goal – 2: Client believes the message in the second round is recognizable.

Client$\mid \equiv\phi (\{(g^{\beta} \bmod p)||h(x||a_i^{/})||(ID_u^{/}||Q_1^{/}||Q_2^{/}||Q_4||Q_5||Q_6||Q_7||sk_s||a_i^{/})||a_i^{/})\}_{Q6})$

Goal – 3: Both Client and Server believe the session share key is recognizable

Client $\mid\equiv\phi$ Server $\mid\equiv\phi$ $(sk_s=sk_u)$

Goal – 4: Client believes server and authenticates server by receiving messages in the second round-trip;

Client $\mid\equiv$ Server$\mid\sim (\{(g^{\beta} \bmod p)||h(x||a_i^{/})||(ID_u^{/}||Q_1^{/}||Q_2^{/}||Q_4||Q_5||Q_6||Q_7||sk_s||a_i^{/})||a_i^{/})\}_{Q6})$

Client $\mid\equiv$ Server$\mid\sim (sk_s=sk_u)$

Goal – 5: Both the Client and Server believes that the share session key among both is

Client $\mid\equiv$ Server $\mid\equiv$ Client $\xleftrightarrow{sku\&sks}$ Server

Table 4. GNY Logic (Odlyzko, A. M, 1984) of Authentication Protocol Steps

GNY PROTOCOL STEPS	DESCRIPTIONS
(M, N)	Conjunction of two formulas M & N
{M}K and {M}K-1	Encryption & Decryption of M with key K
H(M)	A one-way Hash Function
*M:M	M is not initiated here
P◁X	P told X
P∋X	P possesses X
P|~X	P conveyed X
P|≡#(X)	P believes that X is fresh
P|≡ϕ(X)	P believes that X is recognizable
$P\|\equiv \overset{s}{\longleftrightarrow} V$	P believes that X is a suitable secret for P & V
P|⇒X	P has jurisdiction over X
P◁*X	P told X that is not previously covey

Gola – 6: Client shares sk_u, while server shares sk_s and both believe each other

Client $|\equiv$ Client$|\equiv \overset{sku}{\longleftrightarrow}$ Server, Server$|\equiv \overset{sks}{\longleftrightarrow}$Client

Goal – 7: Server also believes that the high entropy random numbers, Multiplicative group G, and group generator g all are fresh.

Server ∋ b_i, Server ∋ G, Server ∋ g, Server ∋ r_i, Server ∋ p and Server ∋ g^a

Goal – 8: Client believes that server possesses sk_u and vice versa

Client$|\equiv$ ∋ sk_u, Server$|\equiv$ ∋ sk_s Client$|\equiv$ ∋ $\overset{sku}{\longleftrightarrow}$ Server, Server$|\equiv$ ∋ $\overset{sks}{\longleftrightarrow}$Client

Goal – 9: Server believes that client believes that sk_u&sk_s at both side considered being shared session key and keeping for secure transferring of sensitive secret information

Server$|\equiv$Client$|\equiv$ sk_u&sk_s

5.3 ProVerif Coding for the Scheme

In the second part of this section, a formal verification of the scheme is performed using programming toolkit ProVerif (Mart´ın & Andrew, 1998). At the beginning two separate channels will be selected, one secret channel sch while the other is open channel ch for the exchanging of data among client and server respectively

```
(*——-channels———-*)
free ch: channel.
free sch: channel [private].
(*——-shared keys——-*)
free sku: bitstring [private].
free sks: bitstring [private].
(*——S's secret key——*)
free x:bitstring [private].
(*—-constants————*)
free IDi:bitstring [private]. (*IDi*)
free PWi:bitstring [private]. (*PWi*)
const g:bitstring. (*generator in G*)
table d(bitstring,bitstring). (*table in S*)
(*——-functions———-*)
fun h(bitstring):bitstring. (*hash function*)
fun h1(bitstring):bitstring. (*hash function*)
fun senc(bitstring,bitstring):bitstring.(*symmetric encryption*)
```

```
fun exp(bitstring,bitstring):bitstring.(*exponent*)
fun xor(bitstring,bitstring):bitstring.
fun con(bitstring,bitstring):bitstring.(*string concatenation*)
fun T1:bitstring
(*--reduction---*)
reduc forall m:bitstring, n:bitstring; sdec(senc(m,n),n)=m.(*symmetric
decryption*)
(*--equations---*)
equation forall m:bitstring,n:bitstring; xor(xor(m,n),n)=m.
equation forall m:bitstring,n:bitstring; exp(exp(g,m),n) =exp(exp(g,n),m).
(*---event----*)
event UserStart(bitstring).
event UserAuth(bitstring).
(*---queries---*)
query attacker(sku).
query attacker(sks).
query id:bitstring; inj-event(UserAuth(id)) ==> inj-event(UserStart(id)).
(*--User's process--*)
let User=
new bi:bitstring;
let HPWi=h(con(PWi,bi)) in
out(sch,(IDi,HPWi));
in(sch,(xZ1:bitstring,xZ2:bitstring,xai:bitstring));
let ai = xai in
let Z1 = xZ1 in
let Z2 = xZ2 in
let Z3 = xor(con(IDi,PWi),bi) in
!
(
event UserStart(IDi);
let bi = xor(Z3,h(con(IDi,PWi))) in
new alpha:bitstring;
new ri:bitstring;
new T1:bitstring;
let HPWi = h(con(PWi,bi)) in
let Ri = xor(Z1,ri) in
let Q1= xor(h(con(IDi,PWi)),ri) in
let Q2= exp(g,alpha) in
let ki = xor(Z2,HPWi) in
let Q3 = h(con(con(con(con(con(IDi,Q2),ki),ri),ai)T1)) in
let P1 = senc(con(con(con(IDi,Q2),Q3),ri),Q1 ) in
let Message1 =(Ri,ai,P,T1) in
out(ch,Message1);
in (ch,Message2:bitstring);
let xQ6 = xor(ki,h(ri)) in
let      (xQ4:bitstring,xQ7:bitstring,xQ8:bitstring,      xxai:bitstring)=
sdec(Message2, xQ6) in
let xQ5 = exp(g,xC4) in
```

```
let sku = h1(con(con(con(con(con(IDi, Q1),Q2),xQ4),xQ5),ki)) in
if                                    xQ8                                    =
h(con(con(con(con(con(con(con(IDi,Q1),Q2),xQ4),xQ5),xQ6),xQ7),sku),xxai
)) then
let xxZ1 = xor(h(con(IDi,HPWi)),xQ7) in
let Z1 = xxZ1 in
let ai = xxai in
0
).
(*------------------Server Process----------------*)
let SReg =
in(sch,(rIDi:bitstring,rHPWi:bitstring));
new ai:bitstring;
new Ni:bitstring;
insert d(rIDi,Ni);
let Z1 = xor(h(con(x,ai)),h(con(rIDi,rHPWi))) in
let Z2 = xor(h(con(con(rIDi,x),Ni)),rHPWi) in
out(sch,(Z1,Z2,ai)).
let SAuth =
in (ch,(xRi:bitstring,xai:bitstring,xP1:bitstring));
let xQ1 = xor (h(con(x,xai)),xRi) in
let (xIDi:bitstring,xQ2:bitstring,xQ3:bitstring,xri:bitstring)=sdec(xP1,xQ1)
in
get d(=xIDi,Ni) in
let xki=h(con(con(xIDi,x),Ni)) in
if xQ3 = h(con(con(con(con(xIDi,xQ2),xki),xri),xai)) then
event UserAuth(xIDi);
new beta:bitstring;
new ai:bitstring;
new T2:bitstring;
let Q4 = exp(g,beta) in
let Q5 = exp(xQ2,beta) in
let Q6 = xor(xki,h(xri)) in
let Q7= xor(x,ai) in
let sks = h1(con(con(con(con(con(xIDi,xQ1),xQ2),Q4),Q5),xki)T1)) in
let                                    Q8                                    =
h(con(con(con(con(con(con(con(xIDi,xQ1),xQ2),Q4),Q5),Q6),Q7),sks),ai))
in
let Message2= senc(con(con(con(Q4,Q7),Q8),ai),Q6), T2 in
out(ch,Message2).
let S = SReg | SAuth.
process !User | !S
```

The above-mentioned program has been executed on ProVerif 1.93. The following result has been displayed.

```
"- - Query inj-event(UserAuth(id)) ==> inj-event(UserStart(id))
Completing...
Starting query inj-event(UserAuth(id)) ==> inj-event(UserStart(id))
RESULT inj-event(UserAuth(id)) ==> inj-event(UserStart(id)) is true.
- -Query not attacker(sks[ ])
Completing...
Starting query not attacker(sks[ ])
RESULT not attacker(sks[ ]) is true.
- -Query not attacker(sku[ ])
Completing...
Starting query not attacker(sku[ ])
RESULT not attacker(sku[ ]) is true."
```

The aforementioned result shows that both the client and server exchange information both at the beginning and ending sessions successfully verify the secret session key not exposed to the adversary. Therefore, the privacy of both peers is well-maintained.

6. Informal Security Analysis

In this section of the article, the focus will be on the proposed scheme which is robust against all known attacks. Let suppose an attacker can interrupt form all routes and modify, copy, replay messages and inject wrong information during the communication. The following some well-known attacks are discussed as an assumption for the scheme.

6.1 Denning-Sacco Attack

Assume that an attacker gets the previous session key sk_s or sk_u, he/she cannot extract user password form it because both the keys are created by large prime numbers and randomly chosen by the client and server respectively. The adversary also can't guess the HPW_i or the server secret session key. In other words, if an attacker replays an old message about session key, he/she cannot get the password from it. Moreover, in each interval, a different session key is created which depends on the client side high entropy random numbers r_i and a_i, the attacker so far, might not compute the session key $sk_s=h1(ID_i^{/}||Q_1^{/}||Q_2^{/}||Q_4||Q_5||k_i^{/})$ and $sk_u=h_1(ID_i||Q_1||Q_2||Q_4^{/}||Q_5^{/}||k_i)$. The proposed authentication scheme, therefore, can resist Denning-Sacco (DS) attack.

6.2 Stolen-Verifier Attack

The authentication scheme presented above has no database for the matching password, so, if an adversary obtained useful information, he/she cannot extract the password from it. The proposed authentication scheme, therefore, strongly resists against stolen-verifier attack.

6.3 Insider Attack

The proposed authentication scheme has no physical database in the server for matching password, even though if an adversary obtains the identity (ID_i) he/she cannot extract password form it. The proposed authentication scheme, therefore, can also resist the insider attack.

6.4 Password Disclosure Attack

In the first phase of the proposed authentication scheme, the client side transmits $\{ID_i, HPW_i\}$ to the server. The client actually does not send the password to the server as in ordinary text format, but it is mixed several times with a random integer values b_i, a_i, r_i and time stamp T. The adversary couldn't find any chance for getting the password at any computation levels. The proposed authentication scheme can, therefore, show resistance to the password disclosure attack.

6.5 Certified-Key Guarantees

The $sk_s=h_1(ID_i^{/}||Q_1^{/}||Q_2^{/}||Q_4||Q_5||k_i^{/})$ and $sk_u=h_1(ID_i||Q_1||Q_2||Q_4^{/}||Q_5^{/}||k_i)$ secret keys are generated depends on the arbitrary numbers chosen by the client say b_i, p and q ; and server side is k_i, r_i and a_i unsystematically and freely in each session. So both the keys must be unique for different sessions, thus, the proposed authentication scheme attempts and guarantees for Certified-Keys for both peers.

6.6 Man-in-the-Middle Attack

After the confirmation of Q_3, both the client and server communicate using session shared keys sk_u and sk_s correspondingly. If an adversary attempts to make its own session with the server, he/she neither shares sk_u nor sk_s,

because the intruders have to calculate and confirm Q_3. Moreover, the adversary might not know the password, identity, and high entropy random secrete numbers k_i, a_i and r_i or the server secret key x. Also, the adversary couldn't guess the server sk_s and Q_3, because it is very difficult for him/her to extract the high entropy random integer numbers b_i and $k_i^{/}=h(ID_i^{/}||x||N_i)$ for calculating Q_3. Therefore, the attacker cannot create its individual connection with server or user. The proposed authentication scheme thus resists man-in-the-middle attack.

6.7 Mutual Authentication

Both the server and user can validate each other by sharing session shared keys sk_u and sk_s respectively, which offers and guarantee for secure mutual authentication.

6.8 Online Password Guessing Attack

The login and authentication computations process is recognized for incomplete determination with wrong password and identity. Afterward the wrong efforts for guessing the password it can automatically block and request the server for interfering to re-activate and unlock. The user's password is also safe along with identity, b_i and high entropy random integer numbers k_i, r_i and a_i. Thus without the knowledge of secret session keys, the attacker, therefore, cannot guess the password of a user online. On another way, if an adversary, for example, efforts for extracting password form Q_3 he/she required a deep knowledge of the random integer number b_i, user identity, and password which is difficult. The proposed authentication scheme, therefore, can strongly resist online password guessing attack.

6.9 Offline Password Guessing Attack

The parameters $\{Z_1, Z_2, Z_3, g, p, a_i$ and $h(.)\}\}$ are stored in the memory of a smart card in the registration phase of the scheme. It cannot only expose to guess by anyone but whenever stolen no one can extract these parameters from it because the discrete logarithmic technique is applied for protecting user identity and password. And also calculated with the random arbitrary number $HPW_i=PW_i \oplus b_i$. The function of XOR operation with password and identity $Z_3=h(ID_i||PW_i)\oplus b_i$ can also protect and difficult to guess. So, guessing of the password it needs to compute 3 unknown arguments which are difficult, therefore, the proposed authentication scheme can resist offline password guessing attack.

6.10 Resist Replay Attack

An attacker if for example intercepts $Message_1= \{R_i, a_i, E_{Q1}(ID_i||Q_2||Q_3||r_i||T_1)\}$ and replays it next time, he/she must be failed to do so because the time makes the identity dynamic which changes every time with the passage of time.. Similarly, if an attacker attempts to replays on $Message_1= \{R_i, a_i, E_{Q1}(ID_i||Q_2||Q_3||r_i||T_1)\}$, he/she requires to exactly compute the high entropy random integer values a_i and r_i. Also, the attacker needs to interrupt parameters from Q_2 but failed due to multiplicative group g and large random number p.

An adversary, let suppose attempts to disturb $Message_2= \{E_{Q6}(Q_4||Q_7||Q_8||a_i^{/}), T_2\}$ and replayed it some other time towards client side, a feasible procedure of dynamic identity has been adopted in the proposed authentication scheme, of which time stamp is embedded for freshness and make it different each time $Q_8= h_1(ID_i^{/}||Q_1^{/}||Q_2^{/}||Q_4||Q_5||Q_6||Q_7||sk_s||a_i^{/}||T_2)$. In the of dynamic identity technique, the user actual identity is concealed in each session called pseudonym. The client after getting the aforesaid message suddenly calculates the timestamp and rejects the irrelevant replay. The proposed authentication scheme, therefore, resists replay attack.

6.11 Strong User Anonymity

A technique of dynamic-ID for the proposed scheme has been adopted in which timestamp T is concatenated with other credentials to make user's actual identity safe from the knowledgeable attacker. The identity for each session is generated differently so that no one trace it during computations due to realistic procedure the dynamic-ID technique $Q_8= h_1(ID_i^{/}||Q_1^{/}||Q_2^{/}||Q_4||Q_5||Q_6||Q_7||sk_s||a_i^{/}||T_2)$ which was first introduce by Das *et al. 's* (2018). Therefore, the proposed authentication scheme showed strong anonymity.

6.12 Resist Denial-of-Service Attack

The proposed authentication scheme not only guarantees for mutual authentication, secret session key but resists replay attack, and offers the services of smart card – an integrated complex circuity and self-computation tool which confirms the legality of a peer. Similarly, the client user gives his/her password and identity, the smart card authenticates the correctness of it, if anyone wrong among these, the smart card suddenly terminate the process because the login and authentication phases parameters depend on server credentials but it could be verified by client due to the availability of smart card. Therefore, the proposed authentication scheme strongly denies DoS attack.

6.13 Common Adversary Model

Suppose a Common Adversary Model (Xu, Zhu, Wen, Jin, Zhang, & He, 2014) is presented here in this research work in which the adversary has full control over the network, has the ability to affect the communication line, can copy, replay, alter, remove messages or can direct false reproduction of messages, can also amend facts, damages useful information on smart card by inquiring via several guesses or get out information and can show itself is a virtual server. Then the proposed authentication scheme can resist all attempts of the adversaries in all routes.

7. Performance Analysis

In this section the performance of the proposed authentication scheme is analyzed by the help of many networking features used by different researchers in the literature as discussed one by one in the following terms:

7.1 Attack Resistance and Functionality Analysis

The attack resistance and functionality analysis of the proposed authentication scheme is compared with Liao *et al.'s* (Liao & Wang, 2009), Chang *et al.'s* (Chang & Lee, 2012), Tsai *et al.'s* (Tsai,Lo, & Wu, 2013), Juang *et al.'s* Juang *et al.'s* (Juang, & Chen, 2008) and Wang *et al.'s* (Ding, & Chun, 2012) schemes, where it can be determined that the proposed user authentication scheme provides resistance to all known attacks which in terms shows robustness, privacy-preserving, and strongly recommended authentication scheme as shown in Table 5 below:

Table 5. The Functionality Comparison

SCHEMES SECURITY PROPERTIES	(Liao & Wang, 2009)	(Chang & Lee, 2012)	(Tsai, Lo, & Wu, 2013)	(Juang, & Chen, 2008)	(Ding, & Chun, 2012)	Proposed
Resists Denning-Sacco-Attack	✓	✓	✖	✖	✖	✓
Resists Stolen-Verifier Attack	✓	✖	✓	✓	✖	✓
Resists Insider Attack	✖	✖	✖	✓	✖	✓
Resists Password Disclosure Attack	✓	✖	✓	✖	✓	✓
Resists Replay Attack	✓	✓	✓	✖	✓	✓
Strong User Anonymity	✖	✓	✓	✓	✖	✓
Rests Server Spoofing Attack	✓	✖	✓	✓	✓	✓
Provides Mutual Authentication	✖	✓	✖	✓	✖	✓
Provides Certified-Key Guarantee	✓	✓	✖	✓	✓	✓
Resists Impersonation Attack	✓	✖	✓	✖	✓	✓

7.2 Storage Overhead Analysis

The smart card is storing $\{Z_1, Z_2, Z_3, g, p, a_i$ and $h(.)\}$ parameters and different key pairs "$b_i, a_i, k_i, \alpha, \beta, p, g$" which occupy 128 and 160 bits key length respectively, and the length of ID_i value is also 160 bits. Therefore, the storage overhead of each participant is listed in Table 6 given below:

Table 6. Storage Overhead

Parameters	**Storage Overhead (in bits)**
Z_1, Z_2, Z_3, g, p, a_i and h(.)	(128+128+128+60+60+60+64)= 628
{k_i, α, β, b_i, p, q}	60x6= 360
User Identity ID_i	160
User Password PW_i	160
Timestamp T	60
Total	**1368**

7.3 Computation Cost Analysis

To check and calculate the computation cost in the eyes of complexity for the proposed authentication scheme, it should be compared with the five latest schemes e.g. Liao *et al.'s* (Liao & Wang, 2009), Chang *et al.'s* (Chang & Lee, 2012), Tsai *et al.'s* (Tsai,Lo, & Wu, 2013), Juang *et al.'s* (Juang, & Chen, 2008) and Wang *et al.'s* (Ding, & Chun, 2012) schemes. The result shows that the proposed authentication scheme is robust, efficient and effective in terms of computational cost. Table 7 demonstrates the assessment.

Table 7. Computational Coast Analysis of Different Schemes

Different Schemes		(Liao& Wang, 2009)	(Chang& Lee, 2012)	(Tsai, Lo,& Wu, 2013)	(Juang,& Chen, 2008)	(Ding,& Chun, 2012)	Proposed
Phases	**Participant**						
Registration	User	0	$1t_{\oplus}+1t_h$	$0+1t_h$	$0+2t_h$	$1t_{\oplus}+1t_h$	$1t_{\oplus}+1t_h$
	Server	$2t_{\oplus}+4t_h$	$7t_{\oplus}+5t_h$	$2t_{\oplus}+3t_h$	$0+4t_h$	$10t_{\oplus}+11t_h$	$2t_{\oplus}+2t_h$
Login	User	$3t_{\oplus}+6t_h$	$6t_{\oplus}+13t_h$	$2t_{\oplus}+8t_h$	$0+2t_h$	$8t_{\oplus}+12t_h$	$3t_{\oplus}+4t_h$
	Server	0	$7t_{\oplus}+19t_h$	0	$0+4t_h$	0	0
Authentication	User	$0+3t_h$	$4t_{\oplus}+3t_h$	$4t_{\oplus}+9t_h$	$0+5t_h$	$0+5t_h$	$2t_{\oplus}+3t_h$
	Server	$3t_{\oplus}+8t_h$	$3t_{\oplus}+2t_h$	$5t_{\oplus}+17t_h$	$0+4t_h$	$6t_{\oplus}+14t_h$	$2t_{\oplus}+6t_h$
Password Change	User	$3t_{\oplus}+2t_h$	s	$3t_{\oplus}+2t_h$	$0+1t_h$		$0+2t_h$
	Server	0		$3t_{\oplus}+2t_h$	$0+3t_h$		0
Card Revocation	User	0		0	0		0
	Server	0		0	0		0
Total for Login and Authentication phases Only		$6t_{\oplus}+17t_h$	$13t_{\oplus}+32t_h$	$13t_{\oplus}+37t_h$	$0+15t_h$	$14t_{\oplus}+31t_h$	$7t_{\oplus}+13t_h$

In the given comparison t_h means the time efficiency of one way hash-function and $t_{\oplus}$ denotes time efficiency for bit-wise X-OR, so at the end result mentioned in the table shows the difference that the proposed authentication scheme time efficiency is less compared to others.

7.4 Communication Cost Analysis

When a legitimate user login into service provider side (server), it is easy to examine that the proposed authentication scheme is somewhere same as Liao *et al.'s* (2009), Chang *et al.'s* (2012), Tsai *et al.'s* (2013), Juang *et al.'s* (2008) and Wang *et al.'s* (2012) schemes, but robust, efficient and effective at the login and authentication phase. Correspondingly, the scheme presented in this article requires a single round-trip to complete, where the other schemes require two to three round-trips for the exchange of message and mutual authentication, respectively. So, the proposed authentication scheme is efficient, simple and effective in enhancing the overall security of client-server architecture.

Suppose the space occupied by each parameter of the proposed authentication scheme is 160 bits, the values for hash function is considered to be 256 bits and the bitwise exclusive-OR operation values always yields 0 bit, then

the whole cost of the proposed authentication scheme at the login and authentication phase can be calculated is in Table 8 below:

Table 8. Communication Cost

	Message	**Communication Overhead/cost (in bits)**
Step 1:	**Message$_1$=** $\{R_i, a_i, E_{Q1}(ID_u\|\|Q_2\|\|Q_3\|\|r_i), T_1\}$	160+60+512+60 = 792
Step 2:	**Message$_2$=** $\{E_{Q6}(Q_4\|\|Q_7\|\|Q_8\|\|a_i^{/}), T_2\}$	512+60 = 572
Total		**1364**

From this, the communication cost of the proposed authentication scheme is somewhat less compared to other schemes in the literature.

8. Conclusion and Future Work

In this research, the discrete logarithmic function is used which is considered to be the basis for its robustness, lightweight and shows a delicate balance between performance and security – because these two are difficult to balance. The key objective was to examine the functionalities of common Discrete Logarithmic Problem (DLP) in a relationship with authentication, tractability, consistency, strength, scalability, security and to focus on the main weaknesses of those schemes based on this method by designing a lightweight, efficient and robust scheme transparently for the client-server architecture. It has also proved that the proposed authentication scheme is the best amongst all in terms of security, tractability, robustness, and lightweight nature. Because the existing schemes are also proficient but have high storage cost, maximum communication, and computation cost and show no balance between performance and security.

In future work, other authentication schemes will be considered for finding out its type, robustness, and methodologies used; so that to adopt a proper mechanism and general framework to keep it secure and minimize the chances of different threats. How and what knowledge or expertise is required to find out or launching an attack on a scheme? The same security authentication scheme will also be extended using Elliptic Curve Cryptography (ECC) and Public Key Infrastructure (PKI) Methods.

References

Bellare, M., Rogaway, P. (1995). Provably secure session key distribution—The third party case, in *Proc. 27th ACM Symp. on Theory of Computing*, ACM, Las Vegas, 1995, pp 57-66. https://doi.org/ 10.1145/225058.225084

Benjamin, C. P., & David, N. T. (2000). *Pict: A programming language based on the pi-calculus. In Gordon Plotkin, Colin Stirling, and Mads Tofte, editors, Proof, Language, and Interaction: Essays in Honour of Robin Milner, Foundations of Computing*. MIT Press, May 2000. https://doi.org/ 10.1.1.127.1777

Blanchet, B., Ben, S., & Vincent, C. (2015). *ProVerif 1.90: Automatic Cryptographic Protocol Verifier, User Manual, and Tutorial*. https://doi.org/ 10.1.1.706.355

Blumenthal, M. S. (2010). Hide and Seek in the Cloud. *Security & Privacy, IEEE, 8*, 57-58. Retreived from http://203.158.98.12/actisjournal/index.php/IJACTIS/article/download/180/105

Burrows, M., Abadi, M., & Needham, R. (1995). A logic of authentication. *ACM Trans Comput Syst, 8*, 108-126. https://en.calameo.com/read/004637821e5086a883f29

Chakraborty, R., Ramireddy, S., Raghu, T. S., & Rao, H. R. (2010). The Information Assurance Practices of Cloud Computing Vendors. *IT Professional, 12*, 29-37. https://doi.org/ 10.1109/MITP.2010.44

Chang, C.-C., & Lee, C.-Y. (2012). A secure single sign-on mechanism for distributed computer networks, *IEEE Trans. on Industrial Electronics, 59*(1), 629-637. https://doi.org/10.1109/TIE.2011.2130500

Chaudhry, S. A., Kim, I. L., Rho, S., Farash, M. S., & Shon, T. (2017). An improved anonymous authentication scheme for distributed mobile cloud computing services. *Cluster Computing*. https://doi.org/10.1007/s10586-017-1088-9

Chuang, M., & Chen. (2014). An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. *Exp Syst Appl, 41*, 1411–1418. https://doi.org/10.1016/j.eswa.2013.08.040

Eric Bach, Discrete logarithms, and factoring. Technical Report UCB/CSD84/186, Computer Science Division (EECS), University of California, Berkeley, June 1984. http://digitalassets.lib.berkeley.edu/techreports/ucb/text/CSD-84-186.pdf

Hassan, A., Eltayieb, N., Elhabob, R., & Li, F. (2017). An efficient certificateless user authentication and key exchange protocol for the client-server environment. *Journal of Ambient Intelligence and Humanized Computing*, 1-15. https://doi.org/10.1007/s12652-017-0622-1

Hsiang, C., & Shih, W. K. (2009). Improvement of the secure dynamic ID-based remote user authentication scheme for a multi-server environment. *Computer Standards & Interfaces, 31*(6), 1118-1123. https://doi.org/10.1016/j.csi.2008.11.002

Hsieh, W., & Leu, J. (2012). Exploiting hash functions to intensify the remote user authentication scheme *Computers & Security*, *31*(6), 791-798. https://doi.org/10.1016/j.cose.2012.06.001

http://prosecco.gforge.inria.fr/personal/bblanche/proverif/. http://dx.doi.org/10.1561/3300000004

Huan, X., Chen, X., Li, J., Xiang, Y., & Xu, L. (2013). Further observations on the smart-card-based password-authenticated key agreement in distributed systems. *IEEE Trans. on Parallel and Distributed Systems, 25*(7), 1767-1775. https://doi.org/ 10.1109/TPDS.2013.230

Jan, S. U. (2017). *An Improved Lightweight Privacy-Preserving Authentication Scheme for SIP-Based-VoIP Using Smart Card*. Anchor Academic Publishing. Retrieved from http://jankp.com/downloads/15-My%20Book.pdf

Jan, S. U., & Fazal, K. (2018). An Improved Forest Fire Alerting System Using Wireless Sensor Network. *Advances in Networks*, 6(1), 21-39, https://doi.org/ 10.11648/j.net.20180601.13

Juang, W. S., Chen, S. T., & Liaw, H. T. (2008). Robust and efficient password-authenticated key agreement using smart cards. *IEEE Trans. Industrial Electronics*, *55*(6), 2551-2556. https://doi.org/10.1109/TIE.2008.921677

Lamport, L. (1981). Password Authentication with Insecure Communication. *ACM Communications*, *24*(11), 770-772. https://doi.org/ 10.1145/358790.358797

Li, C. T., Lee, C. C., Liu, C. J., & Lee, C.W. (2011). A robust remote user authentication scheme against smart card security breach. *25th Annual IFIP WG 11.3 Conference*, p.231-238. https://doi.org/10.1007/978-3-642-22348-8_18

Li, X., Qiu, W., Zheng, D., Chen, K., & Li, J. (2010). Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards. *IEEE Trans. on Industrial Electronics*, *57*(2), 793-800. https://doi.org/10.1109/TIE.2009.2028351

Li, X., Xiong, Y., Ma, J., & Wang, W. (2012). An efficient and security dynamic identity-based authentication protocol for multi-server architecture using smart cards. *Journal of Network and Computer Applications*, *35*(2), 763-769. https://doi.org/10.1016/j.jnca.2011.11.009

Liao, Y. P., & Wang, S. S. (2009). A secure dynamic ID-based remote user authentication scheme for a multi-server environment. *Computer Standards & Interfaces, 31*(1), 24-29. https://doi.org/10.1016/j.csi.2007.10.007

Mart´ın, A., & Andrew, D. G. (1998). A calculus for cryptographic protocols: The pi-calculus. *Information and Computation, 148*(1), 1-70, January 1999. An extended version appeared as Digital Equipment Corporation Systems Research Center report No. 149, January 1998. https://doi.org/10.1006/inco.1998.2740

Miller, H. G., & Veiga, J. (2010). Cloud Computing: Will Commodity Services Benefit Users Long Term? *IT Professional, 11*, 29-37. https://doi.org/ 10.1109/MITP.2009.117

Odlyzko, A. M. (1984). Discrete logarithms in finite fields and their cryptographic significance." *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, Berlin, Heidelberg, 1984. https://doi.org/10.1007/3-540-39757-4_20

Reddy, A. G., Das, A. K., Odelu, V., Ahmad, A., & Shin, J. S. (2018). A Privacy-Preserving three-factor authenticated key agreement protocol for the client-server environment. *Journal of Ambient Intelligence and Humanized Computing*, 1-20. https://doi.org/10.1007/s12652-018-0716-4

Shoup, V., & Rubin, A. (1996). Session key distribution using smartcards", in: *Proc. EUROCRYPT 96*, in: LNCS., vol 1070, Springer-Verlag, pp.321-333, 1996. https://doi.org/10.1007/3-540-68339-9_28

Sood, S. K., Sarje, A. K., Singh, K. (2011). A secure dynamic identity-based authentication protocol for multi-server architecture. *Journal of Network and Computer Applications*, *34*(2), 609-618. https://doi.org/10.1016/j.jnca.2010.11.011

Stallings, W. (2003). *Cryptography and network security: principles and practices (*3th ed.). Prentice Hall. Retrieved from www.inf.ufsc.br/...1/Stallings/Stallings_Cryptography_and_Network_Security.pdf

Sun, D. Z., Huai, J. P., Sun, J. Z., Zhang, J. W., & Feng, Z. Y. 2009. Improvements of Juang *et al.*'s password-authenticated key agreement scheme using smart cards. *IEEE Trans. on Industrial Electronics*, *56*(6), 2284-2291. https://doi.org./ 10.1109/TIE.2009.2016508

Tsai, J.-L., Lo, N.-W., & Wu, T.-C. (2013). Novel anonymous authentication scheme using smart cards. *IEEE Trans. On Industrial Informatics, 9*(4), 2004-2013. https://doi.org/10.1109/TII.2012.2230639.

Wang, D., & Ma, C. (2012). Cryptanalysis and security enhancement of a remote user authentication scheme using smart cards. *The Journal of China Universities of Posts and Telecommunications, 19*(5), 104-114. https://doi.org/10.1016/S1005-8885(11)60307-5

Wang, D., & Wang, P. (2013). Offline dictionary attack on password authentication schemes using smart cards. *16th Information Security Conference*. https://doi.org/10.1007/978-3-319-27659-5_16

Wang, D., & Wang, P. (2014). On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle, and solutions. *Computer Networks*, *73*, 41-57. https://doi.org/10.1016/j.comnet.2014.07.010

Wang, D., Ma, C., & Wang, P. (2012b). Secure password-based remote user authentication scheme with non-tamper resistant smart cards. *26th Ann. IFIP Conf. on Data and Applications Security and Privacy*, p.114-121. https://doi.org/10.1007/978-3-642-31540-4_9

Wang, D., Ma, C., Wang, P., & Chen, Z. (2012). Privacy-preserving two-factor authentication scheme against smart card loss problem. *Journal of Computer and System Sciences*. https://doi.org/10.1007/s11227-015-1610-x

Wang, Y. (2012). Password protected smart card and memory stick authentication against off-Line dictionary attacks. *27th IFIP TC 11 Information Security and Privacy Conference* (pp. 489-500). https://doi.org/10.1007/978-3-642-30436-1_40

Wu, F., Xu, L., Kumari, S., Xiong, L., & Abdulhameed, A. (2015). A new authenticated key agreement scheme based on smart cards providing user anonymity with formal proof. *Security and Communication Networks, 8*(18), 3847-3863. https://doi.org/10.1002/sec.1305

Xu, X., Zhu, P., Wen, Q. Y., Jin, Z. P., Zhang, H., & He, L. (2014). A secure and efficient authentication and key agreement scheme based on ECC for the telecare medicine information system. *J. Med. Syst., 38*(1), 1-7. https://doi.org/10.1007/s13369-017-2665-1

Yang, G., Wong, D. S., Wang, H., & Deng, X. (2008). Two-factor mutual authentication based on smart cards and passwords. *Journal of Computer and System Sciences, 74*, 1160-1172. https://doi.org/10.1016/j.jcss.2008.04.002