Solution to Security and Secrecy in Cloud Environment using PAKE Protocol - A Bibliographic Survey

Anitha Kumari K
Assistant Professor
PSG College of Technology
Coimbatore, India

Sudha Sadasivam G
Professor
PSG College of Technology
Coimbatore, India

Madhumitha R
PG Scholar
PSG College of Technology
Coimbatore, India

ABSTRACT
Cloud computing is an emerging technology that provide people a way to share large amount of hardware and software resources that belong to different organizations. Maintaining privacy and security in cloud environment is more complicated. Mitigating threats in a distributed computing environment is a difficult task as it requires constant vigilance and defense-in-depth. Most systems lack a secure model that guarantees an end-to-end security and confidentiality. Adopting a cloud computing paradigm may have positive as well as negative effects on authenticating a user and exchanging the data. This paper presents an extensive survey of various Password Authenticated Key Exchange (PAKE) protocols available in the literature to eliminate the drawbacks in the current authentication procedure of cloud computing environment.

Keywords
Authentication, cloud computing, PAKE protocol, two server models

1. INTRODUCTION
Cloud computing is a model for enabling ubiquitous, on-demand network access to a shared pool of adjustable computing resources such as networks, servers, storage, applications and services. These resources can be rapidly provisioned and released with minimal management effort. End users access cloud based applications through a web browser using a light weight desktop or mobile applications while the business software and data reside in the remote servers. The main problem of cloud computing is management of the public concerns such as the confidentiality and privacy issues. In order to make a secure usage of the services provided by the cloud, cloud authentication systems can use different password techniques like: i) Simple text password ii) Graphical password and iii) 3D password object. But each of this has its own drawbacks. The weakness of textual password authentication system is that, it is easy to break and it is very much vulnerable to dictionary or brute force attacks. Graphical passwords require memory space greater than or equal to that of the textual password. However, some of the graphical password schemes take a long time. Thus, they are also constrained by time complexity. Similarly 3D-password authentication has its own limitations. To overcome the weakness of the aforementioned protocols and to resist against vulnerable attacks, this paper presents a widespread survey on Password Authenticated Key Exchange (PAKE) protocols and its application in large scale distributed environments. PAKE is method to establish a secret key between two communicating parties based upon their knowledge of a password. Established secret key can be used to secure exchange of messages such that an unauthorized party can obtain no information regarding the messages exchanged without the knowledge of the secret key. This means that if an attacker hacks the server data, still will not be possible to masquerade as the client unless they perform a brute force search for the password.

2. CLOUD AUTHENTICATION PROTOCOLS - AN ANALYSIS
As enterprises changeover to the cloud, it is very essential to control the security needs by means of strong authentication. When data and applications move to the cloud, user access - by default - takes place remotely. As a result, there is a need to transmit data in a secure manner to put back the values into the proper hands. To contend with the complexity of these security and management challenges, there is a need of well-built authentication and key exchange algorithms. Therefore this section analyzes the pros and cons of existing cloud authentication algorithms.

2.1 Graphical Password Based Authentication
Password Authentication System (PAS) [1] for Cloud Environment uses graphical passwords. Graphical-based password technique is developed as a potential alternative to text-based techniques, supported partially by the fact that humans can remember images better than text. Psychologists have confirmed that images are more memorable and usable than text. On the other hand, it is also complex to break graphical passwords using normal attacks such as dictionary attack, brute force and spyware which have been affecting text-based and token-based authentication. Thus, the security level of graphical based validation schemes is higher than other authentication techniques. In this protocol the verification information is absolutely accessible to the user. If the user "clicks" the image for verification and it is compared with the server, the user is implicitly genuine. No password information is exchanged between the client and the server by using PAS authentication system, since the authentication information is conveyed absolutely. Strength of PAS lies in creating a good verification space with adequately huge set of images to shun short repeating cycles. Graphical passwords have memory space which is found to be greater than or equal to the textual password space. But since this particular technique is based on the idea that a picture speaks

thousand words, it is preferred over the text based techniques. However, the graphical password scheme requires a long time for execution. Thus, they are also constrained by time complexity.

2.2 Multidimensional Password Based Authentication

Multi-Dimensional Password Generation Technique for Accessing Cloud Services [2] considers multiple input parameters of cloud paradigm referred as a multidimensional password. The multi-dimensional password is generated by considering the parameters of cloud paradigm such as: vendor details, consumer details, services, privileges and confidential inputs such as logos, images, textual information and signatures. All these dimension combined together produces a multidimensional password. By doing so, the probability of brute force attack for breaking the password can be reduced to a large extent. It is proved that reduction of probability of hacking, improves drastically with increase in dimension of input. However, based on the level of security requirements one can decide the dimension for the input. Major concern is that the processing time increases with increase in dimension of input parameters.

2.3 Textual Password Based Authentication

In textual based password authentication [3] users need not register their passwords to service provider. The Users are supplied with the necessary credential information from the data owner. Furthermore, for enabling the service provider to know the authorized users, data owner provides the service provider with some secret identity information that is derived from the pair (username/password) of each user. The protocol consists of three stages setup, registration, and authentication. Setup and registration stages are executed only once, and the authentication stage is executed whenever a user wishes to login. In the setup and registration stages, the user registers her/his identity (username and password) with Data Owner. Data Owner then provides public system parameters to service provider and each user in secure channel. This scheme is secure against impersonation attack, off-line guessing attack, man-in-the-middle attack, and supports mutual authentication.

2.4 Identity Based Authentication

Identity-based hierarchical model (IBHM) [4] for cloud computing is composed of three levels. The top level (level-0) is root private key generator (PKG). The level-1 is sub-PKGs. Each node in level-1 corresponds to a data-center (such as a Cloud Storage Service Provider) in the cloud computing. The bottom level (level-2) are users in the cloud computing. In identity based hierarchal model of cloud computing (IBHMCC), each node has a unique name. The name is the node's registered distinguished name (DN) when the node joins the cloud storage service. The identity of node is the DN string from the root node to the current node itself. The deployment of IBHMCC needs two modules namely, Root PKG setup and lower level setup which provides secret keys to all nodes. For authentication the client C sends the server S a ClientHello message which contains a random number C_n, session identifier ID and $specification_c$. Then the server sends a ServerHello message that contains another random number S_n, session identifier ID and the cipher $specification_s$. Then C chooses a pre-master secret F_{CS} and encrypts it with the public key P_C of entity C using the encryption algorithm. The cipher text is transmitted to C as ServerKeyExchange message. Then S generates a signature as the Identity Verify message to forward to C. In step (3), C verifies the signature with the help of ID_S. Verification confirms that S is the valid owner of ID_S. This completes authentication form S to which is shown in figure 1.

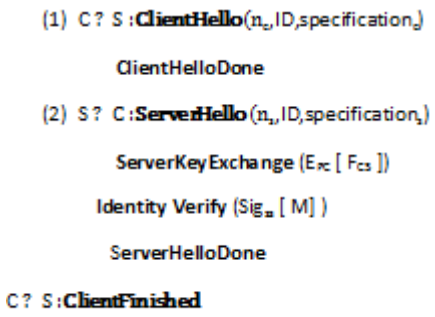

Fig 1: Identity based authentication protocol proposed by Hangwei et al

Then C decrypts the $E_{PC}[F_{CS}]$ with its private key S_C. Because of the fresh F_{CS}, the correct decryption indicates C is the valid owner of ID_C. This step authenticates the validity of C. Eventually, a shared secret key between C and S is generated by K_{CS} = PRF (F_{CS}, n_C, n_S), where PRF is pseudo-random function. Being certificate-free, the authentication protocol aligns well with demands of cloud computing. The authentication protocol is more efficient and lightweight than SLL authentication protocol (SAP), especially for the more lightweight user side. This aligns well with the idea of cloud computing to allow the users with an average or low-end platform to outsource their computational tasks to more powerful servers.

2.5 Biometric Based Authentication

A biometric authentication as a service on cloud [5] uses Single Sign On/Off (SSO) property for authentication. SSO is a property of access control of multiple related, but independent software systems. With this property a user logs in once and gains access to all systems without being prompted to log in again at each of them. Conversely, Single sign-off is the property whereby a single action of signing out terminates access to multiple software systems. A Hybrid biometric method is developed by fusing finger prints and voice biometric methods. The fused value can be used as signal sign on for multiple resources provided by cloud. This encrypted data is used for authentication. The user from an organization uses biometric authentication to connect to the required cloud. The authentication service provider maintains the biometric data base which stores the biometric data in encrypted format. The blind protocol technique [5] reveals only the user identity. As the protocol is based on asymmetric encryption of the biometric data, it captures the advantages of biometric authentication as well as the security of public key cryptography. During the registration process, the user enrolls with the biometric system which is provided by a cloud, once the identity is registered his/her biometric authentication details are stored in cloud service provider database. The authorization details are also entered at the registration time which is then stored in encrypted format. Once authenticated, the user is

redirected to the actual cloud service for which he is authorized to use.

2.6 Three Dimensional (3D) Password Based Authentication

3D password authentication system [6] combines Recognition, Recall, Tokens and Biometrics in one authentication system. The 3-D password is a multifactor authentication scheme. It can combine all existing authentication schemes into a single 3-D virtual environment. This 3-D virtual environment contains several objects or items with which the user can interact. The type of interaction varies from one item to another. The 3-D password is constructed by observing the actions and interactions of the user and by observing the sequences of such actions. The user has the flexibility of selecting the type of authentication techniques that will be the part of their 3-D password. This is achieved through interacting with the objects that acquire information that the user is comfortable in providing and ignoring the objects that request information that the user prefers not to provide. For example, if an item requests an iris scan and the user is not comfortable in providing such information, the user simply avoids interacting with that item. Moreover, giving the user the freedom of choice as to what type of authentication schemes will be part of their 3-D password and given the large number of objects and items in the environment, the number of possible 3D passwords will increase. Thus, it becomes much more difficult for the attacker to guess the user's 3-D password. The strength of this includes unlimited passwords possibility, ease to remember and maintenance of user's privacy. This protocol is secured against the brute force attack and shoulder surfing attack. Table 1 summarizes various authentication protocols applicable to cloud environment.

Table 1. Comparative analysis of cloud authentication protocols

Cloud Authentication Protocol	**Merits**	**Demerits**
Graphical Password Based Authentication[1]	- Images are more memorable and usable than text. - Complex to break graphical passwords using normal attacks such as dictionary attack, brute force	- Larger memory space is required - Higher time complexity
Multidimensional Password Based Authentication [2]	- The probability of brute force attack for breaking the password is reduced to a large extent.	- Processing time increases with increase in dimension of input parameters
Textual Password Based Authentication [3]	- secured against impersonation attack, off-line guessing attack, man-in-the-middle attack, and supports mutual authentication	- Vulnerable to online password guessing attacks
Biometric Based Authentication [4]	- uses Single Sign On/Off (SSO) property for authentication with which a user logs in once and gains access to all systems without being prompted to log in again at each of them.	- Lower user acceptance
Three Dimensional (3D) Password Based Authentication [5]	- Unlimited passwords possibility, ease to remember and maintenance of user's privacy. - Secured against the brute force attack and shoulder surfing attack.	- Higher time complexity

To overcome the weakness of the aforesaid protocols and to minimize the complexities in cloud environment, secure and efficient password authenticated key exchange (PAKE) protocol is required. PAKE ensures secure communication among the servers as well as between user and servers in cloud environment. To achieve this objective, in this paper, a wide survey is carried out on various PAKE protocols. This paper analyzes the pros and cons of various key exchange protocols in single server, multi-server and two server environments. Registration, authentication and key exchange between the users are the basic functionalities of the PAKE protocol as shown in figure 2.

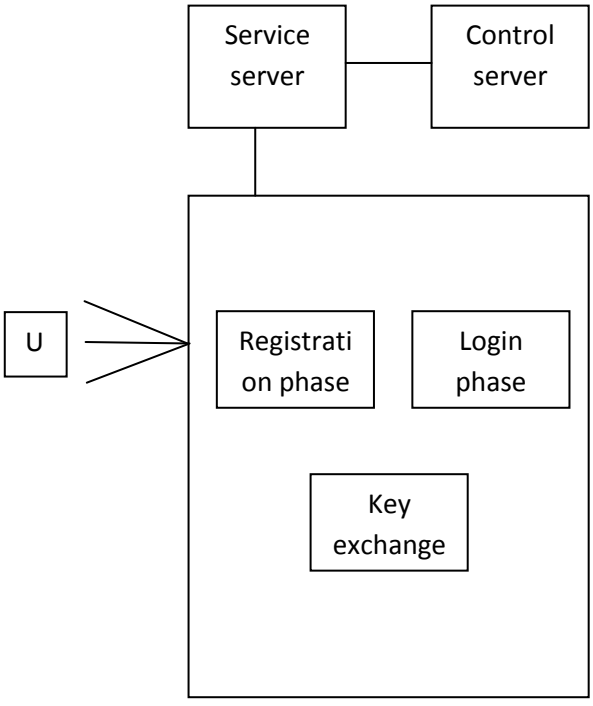

Fig 2: Block diagram of PAKE protocol

In any password system, to enroll as a legitimate user in a service, a user must beforehand register with the service

provider by establishing a shared password with the provider. Authentication phase is where both the servers and client mutually authenticate each other and establish a shared secret key to secure their further communications.

3. SINGLE AND MULTI SERVER AUTHENTICATION AND KEY EXCHANGE PROTOCOLS

Higher-level protocols for authentication and key exchange are frequently developed and analyzed assuming the existence of \secure channels" between all parties, yet this assumption cannot be realized without a secure mechanism for implementing such channels using previously-shared information. The importance of secure key exchange was recognized as early as the seminal work of Diffie and Hellman, which shows how two parties can share a cryptographically-strong key that remains hidden from any passive eavesdropper. The Diffie-Hellman protocol, however, does not provide any form of authentication (i.e., a guarantee that the intended partners are sharing the key with each other), and in particular it does not protect against an active adversary who may inject messages, impersonate one (or both) of the parties, or otherwise control the communication in the network. Achieving any form of authentication inherently requires some information to be shared between the communicating parties in advance of the authentication stage. Historically, authentication protocols were designed under the assumption that the shared information takes the form of high-entropy cryptographic keys: either a secret key which can be used for symmetric-key encryption or message authentication, or public keys (exchanged by the parties, while the corresponding private keys are kept secret) which can be used for public-key encryption or digital signatures. Extensive attention has been given to the problem of designing two-party authentication and authenticated key-exchange protocols under such assumptions and a number of provably-secure protocols relying on shared cryptographic keys are known and described below.

3.1 Encrypted Key Exchange

The first successful password-authenticated key exchange method was encrypted key exchange (EKE) methods [7]. The key exchange between two users is done by a sequence of message exchanges involving both the symmetric and asymmetric encryption. An important concern is the possibility of replay attacks. That is, an attacker with control of the communication channel may insert old, stale messages. Bellovin and Michael's protocol incorporate safeguards, typically in the form of random challenges. The challenge response system is used for validating cryptographic keys. A party sends challenge c encrypted by R, where c was never used before. If the party receives another encrypted message containing c in reply, it follows that the message originator has the ability to encrypt messages with R. Bellovin and Michael's protocol is implemented using RSA. A password P is shared between User A and B in person or by some trusted third party. User A then selects a RSA public key pair E_A which consist of e and n and encrypts e using the shared password P to obtain P(e) and transmits < P(e),n > to the user B along with the identity of A. Suppose if an adversary modifies n as some n`. Then the resulting message from B will be of the form $(R, C_B)^e$ mod n` which is the encrypted version of secrete key R and the challenge C_B generated by B. Now, from a candidate password P` the adversary can compute e`=P`(P(e)). Assuming the adversary knows the factorization of n` the corresponding private key d` is easily computed and can be used to decrypt $(R,C_B)^e$ mod n` obtaining $(R,C_B)^{ed`}$ mod n`. If e is not equal to e`, then this random number, but so is (R,C_B). So the dictionary attack is of no help at this point, and the adversary must still deliver a message of the form $R(C_A, C_B)$ but knows neither C_B nor R. Unable to do so, the attack stops at this point and alarms will ring at both A and B. Thus a novel protocol relying on the counter-intuitive notion of using a secret key to encrypt a public key was developed. Main goal was to protect users with weak passwords without being threatened by dictionary attacks.

3.2 Three Party Encrypted Key Exchange (3PEKE)

A three party EKE protocol [8] allows all clients to share a password with a trusted server which helps two communication parties to mutually authentication each other. The three-party EKE protocol is particularly well-suited for applications that require secure communication between many light-weight and mobile clients. But the difficulties in deploying 3PEKE protocol are, it is impractical to share a common secret between every pair of clients and the requirement of a public key infrastructure, which is often not tolerable. 3PEKE is not resistant to undetectable on-line password guessing attacks [9]. Password guessing attacks can be categorized as Detectable on-line password guessing attacks (use of a guessed password in an on-line transaction), Undetectable on-line password guessing attacks (server does not distinguish an honest request from a malicious one) and Off-line password guessing attacks (attacker guesses a password and verifies his guess off-line). Participation of Server is not required, so Server does not notice the attack. Among the three categories of attacks, off-line password guessing attacks are the most promising ones for an attacker. A secure protocol should ideally resist both types of undetectable attacks. Detectable on-line password guessing attacks can be handled appropriately by introducing exponentially increasing delays after failed attempts and locking the account after an excessive amount of failures.

In 3PEKE protocol (here after referred to as LSH-3PEKE) [10] the server holds a public-key to prevent both off-line and undetectable on-line password guessing attacks. This approach is suitable when the number of message exchanged is of most concern. The drawback of this approach is the burden on user to obtain and verify the public-key of the server [11]. Traditional three-party key distribution services such as Kerberos are susceptible to dictionary attacks with weak passwords and do not provide forward-security.

3.3 Hierarchical Group Password Authenticated Key Exchange (nPAKE$^+$)

In Hierarchical Group Password-Authenticated Key Exchange Protocol Using Different Passwords termed as nPAKE$^+$ the client shares an independent password with a trusted server [11]. The protocol achieves group key establishment and authentication with 3 message flows. The protocol is a combination of hierarchical key tree structure and the password based Deffie Hellman (DH) key exchange. DH Key tree used in this protocol is a binary tree in which each leaf node represents a client. The interior

nodes are not associated with any group member. Each node is associated with a secret key and a blinded key. Group key is public only for that group and it is the secret key of the root node. Initial assumption is that each client shares a password with the server S. First the flow starts from the client C_1, by C_1 sending the request $\{C_i\}^n_{i=1}|X_1^*$ to the next client C_2 where X_1^* is the encrypted version of X_1. The request traverses all clients from C_1 to C_n until it reaches the server. Thus the request to the server consists of n identities and n encrypted exponentials. The second message flow runs in the reverse direction from server S to C_1. The Server decrypts each X_i^* to obtain X_i and chooses s_i for each client C_i to compute the session key $K_i = (X_i)^{si}$. The server then computes Y_i, π and $\tau_i = H(\pi|X_i|Y_i|K_i)$, and sends $\pi|\{Y_i^*|\tau_i\}$ to C_n using which each client can compute their session key K_i after verifying the validity of π. On successful verification of reply, C_i computes R_i, K_i and π, and computes $SK_{i-1} = (BK_{i-1})Ki$ and sends $\pi|\{Y_i|\tau_i\}|R_{i-1}|\xi_{i-1}$ to C_{i-1}. C1 computes the group key GK1 with R1 and K1 as well as π. Then C1 starts the last message flow. Each client $C_i(i = 2, 3, ..., n)$ verifies all the parameters. On successful verification, the client computes the group key GK_i with K_i, L_i,R_i and π. The protocol is secure against the dictionary attacks as the adversary's advantage against the protocol is constrained by the number of send-queries, which represents the number of interactions with a client or a server.

3.4 Gateway-Oriented Password Based Authenticated Key Exchange (GPAKE)

A gateway-oriented password based authenticated key exchange (GPAKE) scheme operates between a client, a gateway, and an authentication server [12]. The authentication server and the client previously share a password for authentication, but a session key is generated between the gateway and the client. The client sends in an encrypted form of information for authentication to the gateway. The gateway forwards the received information to the authentication server and gets back the result of authentication function from the authentication server. The main security goal of the GPAKE scheme is to securely generate a session key between the client and the gateway without leaking information about the password to the gateway. This scheme provides no authentication of message from the client to the server through the gateway and thus it is susceptible to an undetectable on-line password guessing attack by a malicious gateway.

To overcome this drawback, A Message Authentication Code (MAC) is created by the client in the modified GPAKE protocol [13] and the client also enables the server to verify the MAC, where a MAC is assumed to be securely shared between client and server. To establish a MAC key, a 2-PAKE scheme is executed between client and server. It is proved that if 2-PAKE is a secure 2-party password-based authenticated key exchange and $MAC_k(\cdot)$ is a secure MAC algorithm such that $MAC_k : \{0, 1\}* \rightarrow \{0, 1\}l$, then the modified GPAKE scheme is secure against undetectable online password guessing attacks.
In NEW GPAKE [14] the client and the server have pre-shared a password pw, and the channel between the gateway and the server is assumed to be authenticated and private. Its security is based on the Computational Diffie-Hellman problem. In this scheme, the server forwards a challenge to the client, and explicitly verifies the client's response; therefore, the server can explicitly authenticate the client's requests and the conventional on-line guessing attack prevention mechanism can be applied. It is also found to be resistant to un-detectable on-line guessing attack. In on-line guessing attacks, the adversary (gateway) should be able to verify its guess using the response from the client or the response from the server. The response from the client includes the computation of ephemeral Diffie-Hellman key which is only known to the client and the server, and the response from the server also involves the secret value; therefore, the gateway has no way to verify its guess. It also proved that this protocol is resistant to off-line guessing attack. Here the adversary should be able to verify its guess using the communications. However, all the communications in this protocol are either random challenge or secret computations involving secret Diffie-Hellman keys; therefore, an attacker has no way to verify its guess and the scheme is secure against off-line guessing attacks.

3.5 Threshold Password Authentication

A Threshold Password Authenticated Key Exchange [15] uses a set of servers with known public keys. User authentication is successful only when certain threshold of servers accepts the authenticity of the user. An attacker will not be able to perform an offline dictionary attack unless threshold number of servers is compromised. The system is found to secure in the random oracle model under the Decision Diffie-Hellman assumption against an attacker who compromises fewer than threshold of servers. This security is achieved by storing a semantically secure encryption of a function of the password at the servers instead of simply storing a hash code of the password. The distribution of secret decryption keys is done using Feldman verifiable secret sharing. By this technique the problem of distributing password authentication information is transformed to a problem of distributing cryptographic keys.

In a Simple Authenticated Key Agreement Protocol the client and server are assumed to share the weak secret(password) in a secure way [16]. They agree upon the generator g and its group Z_p^*. And x and y are selected in Z_p^* for a uniform distribution, and $X = g^x \bmod p$ and $Y = g^y \bmod p$ are also in Z_p^* for a uniform distribution. The session key is made by $h(g^{xy} \bmod p)$. The protocol satisfies the property of perfect forward secrecy. In perfect forward secrecy an exposed password cannot enable an attacker to derive session keys of past communication sessions. In this protocol, the security of perfect forward secrecy is based upon the Computational Deffie Hellman (CDH) assumption. Even if the attacker knew the correct password pw, the attacker still cannot compute the previous session keys without violating the CDH assumption. It is also found to be computationally efficient because it does not involve any encryption/decryption techniques. Table2 summarizes the various key exchange protocols applicable to single server and multi-server environment. Drawback of these single server environments is that there exists is a credential weakness that a user's password table at the server can be stolen by an adversary. There exists a single point of vulnerability. To eliminate this single point of vulnerability multiple server based password authentication and key exchange systems were proposed. The principle is distributing the password database as well as authentication function to multiple servers so that an attacker has to compromise several servers to be successful in offline dictionary attacks. While the protocols are theoretically

significant, they have low efficiency and high operational complexity. Multiserver password based systems can be broadly classified into two types where all the servers are equally exposed to the users and the user must communicate in parallel with all servers or a gateway is introduced between the client and multiple servers. The main drawback of exposing all servers to the user is the demand on communication bandwidth and the need for synchronization at the user side. In the gateway augmented multi-server model a gateway is positioned between the users and the servers and a user only needs to contact the gateway. Introduction of the gateway removes the drawback of the need for synchronization at the user side to communicate with all servers. However, the gateway introduces an additional layer which is redundant since the purpose of the gateway is simply to relay messages between users and servers and it does not involve in any authentication service. From security perspective, more components generally imply more points of vulnerability. Table 2 gives a comparative analysis of single server and multi-server protocols.

Table 2. Comparative analysis of single server and multi server authentication and key exchange protocols

Protocol	**Merits**	**Limitations**
Encrypted Key Exchange [7]	- resistant against replay attack and online dictionary attack	- Secured when encryption of random secret key by random public key when it leaks no information related either about the secret key or public key
3PEKE [8]	- resistant to online password guessing attacks - provides forward secrecy	- impractical that every two clients share a separate secret key - not resistant to undetectable online guessing attacks - not resistant to offline password guessing attack
LSH-3PEKE [10]	- resistant to offline password guessing attack and undetectable online guessing attack	- Burden on communication parties because they have to obtain and verify the public-key of the server
nPAKE^{+} [11]	- resistant to detectable online and offline dictionary attacks - Mutual authentication is achieved within 3 message flows	- Computationally complex operations
GPAKE [12]	- resistant to offline password guessing attack	- susceptible to an undetectable on-line password guessing attack by a malicious gateway
The modified GPAKE [13]	- secured against undetectable online password guessing attack - Resistant to offline password guessing attack.	- Requires a 2-PAKE scheme to executed between the client and server to exchange MAC key which is later used to provide authentication
Threshold Password Authenticated Key Exchange [15]	- secure against an attacker who may eavesdrop on, insert, delete, or modify messages between the user and servers, and that compromises fewer than that threshold of servers	- Deciding a threshold value is a complex operation.
Simple Authenticated Key Agreement Protocol [16]	- satisfies the property of perfect forward secrecy	- susceptible to offline password guessing attacks
New GPAKE [14]	- resistant to undetectable online password guessing attacks and offline password guessing attacks	- Communication channel between the gateway and server is assumed to be authenticated and private.

Table 3 shows the comparison of single server, multi-server and two server models with respect to complexity, feasibility, vulnerability and deployment strategy.

Table 3. Comparative analysis of single server, multi-server and two server models

Parameters	**Single Server Model**	**Multi Server Model**	**Two Server Model**
Communication Complexity	Very Less	More	Less
Engineering and Economic Feasibility	Yes	No	Yes
Most Deployable	Yes	No	Yes
Single Point of Vulnerability	Prone to vulnerability	Resilience against vulnerability	Resilience against vulnerability
Offline Dictionary Attack	Prone to attacks	Resist against attacks	Resist against attacks

From table 3 it is perceived that to overcome the drawbacks transpiring in single server and multi-server environment, two server authentication models can be used. Different levels of trust can be set upon two servers and usually the back-end server is more trustworthy than the public server. This sounds good, since the back-end server is located in the back-end and is hidden from the public, and it is thus less liable to be attacked.

4. TWO SERVER AUTHENTICATION AND KEY EXCHANGE PROTOCOLS

This section presents a discussion on various two server authentication protocols.

Preliminaries used are:

- π -> user's password
- b1,b2 -> random number
- Q, p, q -> large prime numbers
- g1 & g2 -> are of order q and discrete logarithms to each other
- g3 -> is of order of p
- h(.) -> cryptographic hash function
- U -> user identity
- SS -> Service Server identity
- CS -> Control Server identity

In 2006 the basic two server model to protect a system against a single point of vulnerability and a practical authenticated key exchange protocol upon the two server model was proposed [17]. Their system involves three entities namely users, a service server (SS) that is a public server and a control server (CS) that is the backend server. Their primary goal is to resist offline dictionary attacks by the two servers, where CS is controlled by passive adversary and SS is controlled by an active adversary. This was achieved by strengthening the user's short password π into two long shares π1 and π2 in such a way that they are no longer subject to offline dictionary attack and distribute them to the two servers. As a result an attacker has to compromise both the servers in order to grab the user's password π. During authentication the user U provides his/her password π to the service server SS which uses its share π1 and takes the assistance of the control server CS which provide its share π2 for user authentication. Once the service server SS and the user U authenticate each other, they negotiate a secret session key to secure their further communications. The protocol is secure against offline dictionary attacks by CS as a passive adversary when it eavesdrops on the communication channels, because CS cannot learn anything on π1. It is also proven that the protocol is secure against offline dictionary attacks by SS as an active adversary as it is not possible for SS to change the parameters and also make CS to authenticate U. As a result, as an active attacker, SS is still not effective in offline dictionary attack.

To overcome the drawbacks of basic model, Yang et al. proposed an improved model [17] by introducing an extra parameter g3 for the purpose of user authentication which is shown in figure 3. Clearly by the removal of the secret channel dose not facilitate outside attackers who have no control on any server to derive the session key used between U and SS and at the same time CS cannot compute the session key shared between U and SS. This protocol is also found to be computationally efficient as the computational complexity (number of exponentials to be calculated) is found to be 9

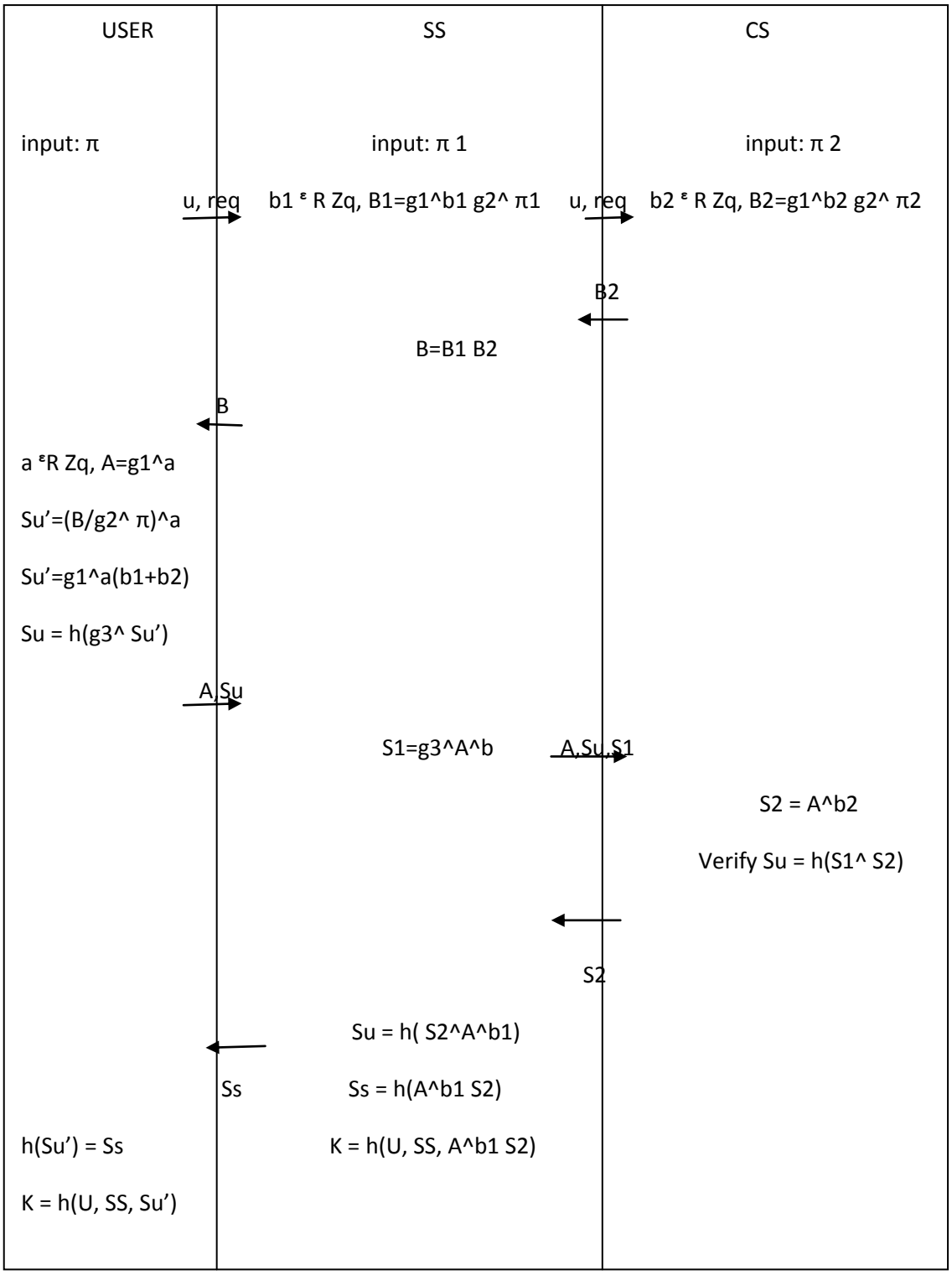

Fig 3: Password authentication and key exchange proposed by Yanjiang et al

where 4 of it can be computed offline. It is also found to be fast as the communicational performance in terms of bits is given by |p| + |h| for Users and CS, 4|p| + 2|h| for SS where |p| and |h| denote the bit length of password p and the hash function h(.), respectively and the communicational performance in terms of rounds is given by 4 for Users and CS, 8 for SS.

A two server authentication and key exchange protocol which support multiple Service Servers SSj and a Single Control server CS [18]. Each service server SSj has its own secret key KSj = h(SSj, x) which is computed by CS. Their protocol is robust against the stolen verification attack without the assumption of deploying a secure database in the service server. The user U must initially register himself with CS using his/her identifier and password. During the authentication phase user U requests the particular SSj with the message < UID || SSj || Req>. The service server SSj calculates its password share πj for the user with the identifier UID and also passes the request to the CS. Once the SSj and CS authenticate each other SSj and U negotiate a secret session key K. While an adversary tries to masquerade as one of the service servers he/she will not make it successful. If one of the legitimate user tries to spoof a server by using his/her knowledge about the server obtained from prior communication with that server, it impossible to succeed without knowing the user password π of another user. If a legitimate server SSi tries spoofs another server SSj, SSi has no knowledge about the password share πj = h(UID || KSj) of SSj. Therefore this protocol is proved to be secured against sever spoofing attack. Since none of the service servers SSj stores any information related to user's password, the protocol is secure against stolen verification attack. This protocol is not efficient when compared with Yang et als protocol because the computational cost is increased by a factor 1 with respect to |h|. But the communicational cost remains to be the same as of [17].

A Novel Two-Server Password Authentication Scheme [19] with Provable Security focuses on how to protect the password information from the compromise of a server and the compromising server does not help an adversary authenticate to the other server. Their protocol is resist against off-line dictionary attacks launched by an active adversary. This protocol has computational complexity (number exponential to be calculated) of 9 where 2 can be computed offline. For the user the computational complexity is reduced from 3 to 1 when compared with Yang et als. protocol in [17]. Whereas communication performance in terms of bits for |p| is increased by a factor of 3 and for |h| it is increased by a factor of 2 where |p| and |h| denote the bit length of password p and the hash function h(.), respectively. But the communicational performance in terms of rounds remains the same as 4 for the User, 8 for SS and 4 for CS when compared with the protocol in [17].

An Efficient password based Two-Server Authentication and Pre-shared Key Exchange System using Smart cards [20] is an ID-based remote user authentication protocol with smart card which uses simple Bitwise XOR operation and one hash function. Their protocol consists of four phases namely, registration, login, authentication, password exchange. During the registration phase, the user registers with the remote server S by providing their identifier and password. The server S computes some parameters related to that user and stores them in the smart card's memory which is then issued to the user. In the login phase a user can gain access rights for a server S by inserting the smart card to card reader and by providing their identity and password. The card reader computes a parameter and compares it with the parameter stored in the card. On successful comparison login request is sent to the Server S. In the verification phase the server S and User mutually authenticate each other. On successful mutual authentication card reader generates a session key K provides it to the User and the server S. This protocol is secured against off-line dictionary attack, replay attack, malicious server attack and man-in-the-middle attack.

A Dynamic Identity Based Authentication Protocol for Two-Server Architecture [21] uses nonce, one-way hash function and XOR operations. In the registration phase, the User has to submit his/her identity and password to the CS which then provides a smart to the user than contains the security parameters related to that user. CS also sends the newly registered user details to the SS. In the Login phase, the card reader verifies the authenticity of the user and sends a request message to the SS with the Nonce N1. In the authentication and session key agreement phase, the SS generates a Nonce N2 and sends the request message to the CS. Once the authenticity of the user and SS is verified by the CS, it generates a Nonce N3 and sends a message back to SS. When the SS verifies the legitimacy of the CS and the card reader verifies the authenticity of both the SS and CS, the user's smart card, SS and CS agree upon on secret session key. This protocol is secured against the malicious server and user attack, stolen smart card attack, replay attack and offline dictionary attack. The communication performance in terms of bits is given by 1152 bits but for other protocols it depends on the size of |p| and |h|. The computational complexity of the user is given by 8TH and for SS and CS it is given by 12TH where TH is time complexity of the hash function. This is found to have lesser computational complexity when compared with other protocols [17], [18], [19], and [20] whose computational complexity is given by the time take for the exponential calculation. A two server password only authenticated key exchange [22] improves the security by encrypting the password using Cramer-Shoup algorithm and servers is provisioned with El Gamal public/secret-key pair. Efficiency gets improved in terms of exponentiations. In client side 15 full exponentiations and meanwhile server side with 13 exponentiations by pre-computation. But this protocol makes use of gateway between client and servers which raise the communicational complexity.

An efficient two server password only authenticated key exchange protocol [23] is a symmetric two server protocol that performs the operations in parallel at both the servers. This protocol makes use of Diffie-Hellman key exchange and Elgamal encryption scheme and is robust against passive and active attacks. This protocol is found to have lesser communication complexity in terms of rounds. To précis effectively, various two server authentication protocols merits and limitations are explicated in table 4.

Table 4. Merits and demerits of two server authentication and key exchange protocols

Protocol	Merits	Limitations
A practical Password based Two-Server Authentication and Key Exchange [17]	-robust against offline dictionary attacks by CS and SS as passive and active adversary respectively. -Secure against an active outside adversary controlling no server.	-Secure communication channel is used for communication between SS and CS. -CS is not robust against dictionary attacks by active adversary. -CS can compute the session key established between U and SS.
Secure and Efficient Password-based Authenticated Key Exchange Protocol for Two-Server Architecture [18].	-secure against server spoofing attacks, server database stolen verified attack -Service servers does not store any information related to user's password in their database	-Unencrypted transfer of user's password to the service server in the authentication phase.
Novel Two-Server Password Authentication Scheme with Provable Security [19]	Secure against off-line dictionary attacks launched by an active adversary.	Secure communication channel is used for communication between SS and CS.
An Efficient password based Two-Server Authentication and Pre-shared Key Exchange System using Smart cards [20].	- Secure against off-line dictionary attack, replay attack, malicious server attack and man-in-the-middle attack.	-Card reader acts as one of the two servers. -Session key is known to the card reader. -Impersonation of card reader could lead insecure communication.
Dynamic Identity Based Authentication Protocol for Two-Server Architecture [21]	-Mutual authentication -Secure against the malicious server and user attack, stolen smart card attack, replay attack and offline dictionary attack	- Use of nonce. - If an expired nonce is used, the server should be able to recognize it.

5. ANALYSIS OF TWO SERVER PAKE PROTOCOLS

The findings in the survey on two server password authenticated key exchange have been analyzed based on key compromise attack, identity theft, number of rounds, feasibility, secrecy, security of session key, offline attack and user anonymity parameters which is exposed below in table 5.

Table 5.Comparative analysis of two server authentication and key exchange protocols

Comparison Parameters	Yang et al	Katz et al	Yi et al	Anamika and Yogadhar	Sandeep
Resilience to Key Compromise Attack	No	Yes	Yes	Yes	Yes
Identity Theft	No	No	No	Yes	Yes
Communication - Number of Rounds	U – 4 S1 – 8 S2 – 4	U-3 S1-6 S2-6	U-3 S1-3 S2-3	U-2 S1-4 S2-2	SC-2 S1-4 S2-2
Economic and Technical Feasibility	Yes	Yes	Yes	No (Device Requirement)	No (Device Requirement)
Forward Secrecy	Yes	Yes	Yes	Yes	Yes
Session Key Security	No	Yes	Yes	Yes	Yes
Resilience to Offline Dictionary Attack	No	Yes	Yes	Yes	Yes
User Anonymity	No	No	No	Yes	Yes
Mutual Authentication	Yes	Yes	Yes	Yes	Yes

6. CONCLUSION

Security is thus a vital research area in distributed and cloud environments. In today's world, it is very important to enhance the security architecture such that the attacks are minimized. Several PAKE protocols have been proposed for single server, multi-server and two server environments in recent years for federated enterprises. In this article we suggested to make use of two server PAKE protocol in large scale distributed environments to eliminate the drawbacks present in existing protocols. Also from table 3

it is apparent that two server model performs better than other models. Each of the two server PAKE protocols in the literature has its own merits and demerits and thus provides effective solutions. From table 3 and 5, it is clear that two server PAKE protocols overcomes the complexities and attacks occurring in other models, since here user credentials are interpreted and stored in two servers. So, there is very less chance to reveal the user credentials to the adversary. From the study, the inference obtained is that even though 100 percent security is not possible by utilizing any protocol, two server PAKE provides better security than others. Thus the security remains in substantial flux.

7. REFERENCES

[1] Bhavana A, Alekhya V, Deepak K, and Sreenivas V, "Password Authentication System (PAS) for Cloud Environment", International Journal of Advanced Computer Science and Information Technology, 2013, Volume 2, pp.29-33.

[2] Dinesha H A, Agrawal V K, "Multi-Dimensional Password Generation Technique for Accessing Cloud Services", International Journal on Cloud Computing: Services and Architecture, 2012, Vol.2, No.3. pp.31.

[3] Ali A. Yassin, Hai Jin, Ayad Ibrahim, Weizhong Qiang, Deqing Zou, "Efficient Password-based Two Factors Authentication in Cloud Computing", International Journal of Security and Its Applications, April, 2012, Vol. 6, No. 2.

[4] Hongwei Li, Yuanshun Dai, Ling Tian, Haomiao Yang, "Identity Based Authentication for Cloud Computing", Springer, First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009, pp. 157-166.

[5] Himabindu Vallabhu, Satyanarayana R V, "Biometric Authentication as a Service on Cloud: Novel Solution", International Journal of Soft Computing and Engineering, September 2012 ISSN: 2231-2307, Volume-2, Issue-4.

[6] Duhan Pooja, Gupta Shilpi, Sangwan Sujata, and Gulati Vinita, "Secured Authentication: 3D Password", International Journal of Engineering and Material Sciences, 2012, VOL.3(2),242 – 24.

[7] Bellovin S M and Merrit M, "Encrypted key exchange: Password based protocols secure against dictionary attacks," Proc. IEEE Symp.on Research in Security and Privacy, 1991, pp. 72–84.

[8] Steiner M, Tsudik G and Waidner M, Refinement and extension of encrypted key exchange," ACM Operating Syst. Rev., 1995, vol.29, no. 3, pp.22–30.

[9] Yun Ding and Patrick Horster, "Undetectable on-line password guessing attacks, "ACM Operating Syst. Rev., 1995, vol. 29, no. 4, pp. 77–86.

[10] Lin, H.-M. Sun, and T. Hwang, "Three-party encrypted key exchange: Attacks and a solution," ACM Operating Syst. Rev., 2000, vol. 34, no.4, pp. 12–20.

[11] Zhiguo Wan, Robert H. Deng, Feng Bao and Bart Preneel,"nPAKE+: A Hierarchical Group Password-Authenticated Key Exchange Protocol Using Different Passwords", ICICS 2007, 2007. LNCS 4861.

[12] Abdalla M, Chevassut O, Fouque, Pointcheval D, "A simple threshold authenticated key exchange from short secrets," Springer-Verlag -in Proc. ASIACRYPT , 2005, LNCS vol. 3788, pp. 566-584,

[13] Jin Wook Byun, Dong Hoon Lee, Jong In Lim, "Security Analysis and Improvement of a Gateway-Oriented Password-Based Authenticated Key Exchange Protocol", IEEE Communications Letters, September 2006, vol.10, no.9, pp.683,685.

[14] Hung-Yu Chien, Tzong-Chen Wu, Ming-Kuei Yeh, "Provably Secure Gateway-Oriented Password-Based Authenticated Key Exchange Protocol Resistant to Password Guessing Attacks", Journal Of Information Science And Engineering 29, 2013, pp. 249-265.

[15] Philip MacKenzie, Thomas Shrimpton, "Threshold Password Authenticated Key Exchange", ACM, Journal of Cryptology, 2006, Vol 19, Issue 1, pp.27-66.

[16] Her-Tyan Yeh, Hung-Min Sun, "Simple Authenticated Key Agreement Protocol Resistant to Password Guessing Attack", Journal Of Information Science And Engineering 19, 2003, pp. 1059-1070.

[17] Yanjiang Yang, Robert H. Deng and Feng Bao, "A practical password-based two server authentication and key exchange system", IEEE Transaction on Dependable and Secure Computing, 2006, 3(2):105–114.

[18] Jun Ho Lee and Dong Hoon Lee, "Secure and Efficient Password-based Authenticated Key Exchange Protocol for Two-Server Architecture", International Conference on Convergence Information Technology, 2007, pp. 2102-21207.

[19] Dexin Yang and Bo Yang, "A Novel Two-Server Password Authentication Scheme with Provable Security", IEEE 10th International Conference on Computer and Information Technology, 2010, pp. 1605-1609.

[20] Anamika Chouksey, Yogadhar Pandey, "An Efficient password based Two-Server Authentication and Pre-shared Key Exchange System using Smart Cards", International Journal of Computer Science and Information Technologies, 2013, Vol. 4 (1) pp.117-120.

[21] Sandeep K.Sood, "Dynamic Identity Based Authentication Protocol for Two-Server Architecture", Journal of Information Security, 2012, pp.326-334.

[22] Jonathan Katz, Philip Mackenzie, Gelareh Taban and Virgil Gligor, "Two-Server Password-Only Authenticated key Exchange", Elsevier – Journal of Computer and System Sciences", March. 2012, Vol 78, Issue 2, pp.651-669.

[23] Yi, Xun; Ling, San; Wang, Huaxiong, "Efficient Two-Server Password-Only Authenticated Key Exchange," IEEE Transactions on Parallel and Distributed Systems , Sept. 2013, vol.24, no.9, pp.1773-1782.