Signature Based Key Exchange for Securing Data and User From Web Data Stealing Attacks

Saravanan Arumugam, Sathya Bama Subramaniam, Kirubanand V B

Abstract: ***Due to the immense technological growth, web and its related applications are becoming a major part of everyday life. The growth of the internet and technology not only increases the positive benefits but also increases negative activities such as data theft. As web applications are used frequently for many online services, it is the most common and valuable target for the adversary to host any web vulnerabilities. Data theft or data stealing attacks are quite common in the web and the internet with severe consequences. The private data are generally stored on the system which gives an opportunity for the attacker to steal the data from the storage or during transit. However, apart from stealing the critical data from the user, the attacker also steals the sensitive data from the web applications. This type of attack takes several forms for stealing perilous information from the user and web application. Unfortunately, these attacks are easy to execute as the attacker needs only the internet connection, a web server and technical knowledge which are readily available. Several prevention strategies exist to thwart the user and the application from the web attacks, however, they do not provide the complete solution. This paper presents the signature based key exchange to prevent the user as well as the web application from several variations of data stealing attacks through mutual attestation. The experimental results show that the proposed method prevents the user and application from data theft than any other existing methods.***

Index Terms: ***Data Theft, Signature based Key Exchange, Mutual Authentication, Data Stealing Attacks, Web Application Securit***

I. INTRODUCTION

The various services provided by the internet have become inevitable in our everyday life and, gains an extensive acceptance. On average, nearly three out of four humans rely on the internet for their everyday activities such as online banking, shopping, online marketing, and communication. These services increase and speed up our daily activities using digital payment and online order options. This positive uses of the internet and the web really make our complicated life as simple. Web application is a program that is hosted on the remote web server. The client normally web browser sends its request through the internet. The server responds to the received client's request.

As the web and the internet have become a promising area for the client to get several services, the malicious attacker mainly focuses on the web application. However, the data stored on the web applications are the most exposed and highly sensitive as it contains privacy data, the consequence of the attack is severe. This makes the researchers turn their head on web security and to focus on the security of web applications and its users by suggesting prevention and detectionn measures.

Thus, the web application security is the major concern in preventing web applications and web data. To protect the web applications from the adversary, a non-profitable organization popularly known as Open Web Application Security Project (OWASP) is urging the web developers and researchers to develop and maintain applications in a trusted way [1]. The organisation also conducts and publishes the top 10 vulnerabilities every year. Accordingly, as the survey made during the year 2017, Injection, Broken authentication, Sensitive data exposure, XML external entities, Broken access control, Security misconfigurations, Cross-site scripting, Insecure deserialization, Using components with known vulnerabilities and Insufficient logging and monitoring are the top 10 vulnerabilities identified [2]. In this list, many of the vulnerabilities belong to data stealing and data theft which cause serious damage to the underlying application.

As the number of applications increases unpredictably, the number and complexity of vulnerabilities also continue to increase dramatically. Numerous developers find difficult to ascertain the bugs and to assess the security of the web application, as the application becomes more complex and large due to the technological advancement. According to WhiteHat Security analyses, many of the web applications are failed to implement the security inside the application. Additionally, 70% of the web application is implemented using reusable components due to the quick and simple development. However, these reusable components may be developed by the third parties and as a result, it might have inherent vulnerabilities [3]. Also, according to the Micro Focus, 79% of the web applications that undergone analyses

Revised Manuscript Received on December 22, 2018.

Saravanan Arumugam ,Department of MCA, Sree Saraswathi Thyagaraja College, Pollachi, Tamil Nadu, India – 642205 a.saravanan21@gmail.com

Sathya Bama Subramaniam,Independent Researcher, 483, Lawley Road, Coimbatore, Tamil Nadu – 641003 ssathya21@gmail.com

Kirubanand V B,Department of Computer Science, Christ University, Hosur Road, Bengaluru – 560029, vbkiruba72@gmail.com

had at least one critical vulnerability [4]. And these vulnerabilities are increasing every year.

From the various report statistics on web vulnerabilities and web application security risks, it is clear that securing the web application and its users are obvious and mandatory. Thus, the proposed method protects the client and server from the several variations of data stealing attacks. The method has several security features where the client credentials are used to secure them from the fake websites by establishing mutual authentication between the client and server. Only after the mutual authentication, the server permits the sensitive data to the client. Also, sensitive data is transmitted over the communication medium only after the successful authentication. The method prevents the web application from the man in the middle attack as the request and response cannot be forwarded as a legitimate user. Thus, the main aim of this work is to shield the web application from several critical security risks in data theft.

The flow of the paper is as follows; Section II describes the various forms of data stealing attacks. Section III presents the literature survey related to the present work. The proposed signature based key exchange method is explained in section IV. Section V presents security analysis of the proposed method based on various policies and resistance against various data stealing attacks and the comparison with other existing techniques. Finally, section VI concludes the paper.

II. DATA STEALING ATTACK

Data stealing attack or simply data theft is a serious cyber-attack wherein a malevolent attacker interrupts and transmits messages between client and server. The main trick played by the attacker is the imitation as client and server for accessing the sensitive information. Once the imitation is successful, several variations of the attack such as code injection, redirection attacks, port stealing will be initiated by the adversary. Phishing and Man-in-the-middle attacks are the base for any data theft in web applications. Phishing websites are another form in which the fake site that runs in phisher's host by impersonating as a legitimate application on a legitimate host and looks very similar in appearance as a legitimate website. This makes the user believe that they are using the trusted website [5].

Man-in-the-middle attack (MITM) is a critical attack where the attacker secretly eavesdrops and in some cases alters the communication that is carried out between two legitimate parties and unfortunately, the legitimate users believe that they are communicating directly to each other [6]. Applications like online banking and shopping normally become the victim of the data stealing attacks and the attacker collects the sensitive credentials and information. Several attacks are being processed by the attacker to steal data. Fig.1 shows the various data stealing attacks in communication between the web user and web application.

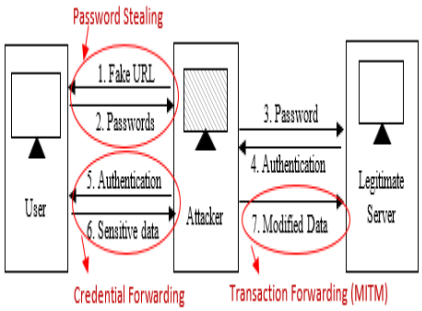


Fig. 1 Various Data Stealing Attacks

Password stealing attacks attempt to steal the password and, then other sensitive information from the user by impersonating as a legitimate web application. Once the details are collected the link between the attacker and the user gets disconnected and the attacker communicates with the application as a legitimate user. However, in some cases, the user may provide an incorrect credential which leads to failure attack. Thus, in credential forwarding attacks, instead of disconnecting the user immediately after acquiring a password, the attacker forwards the credentials to the legitimate application to verify the authenticity.

On successful authentication, the attacker then provides the fake URL and collects the sensitive information from the user imitating as the legitimate web application. Another data stealing attack is transaction forwarding attack which is similar to the credential forwarding attack. The main difference is that instead of sending the fake URL to the legitimate user, the attacker captures the data, modifies the request sent by the legitimate user to the web application. The application normally accepts the modified data as if it comes from a legitimate user.

Any security system must have the three main security policies such as Confidentiality, Authenticity and Integrity [7]. Thus, the developer must implement the system in such a way that it must satisfy the following design goals [8].

1. Only legitimate communication parties (client and a server) should access sensitive data.
2. Other untrusted input should not be served to the underlying web application.

Based on the two design goals, the signature based key exchange has been proposed that secure the communication using mutual attestation from various data stealing attacks.

III. RELATED WORK

For any secure and private communication over an insecure medium, there is an indispensable need for the secret keys to be exchanged between the two communication parties. The Password Authenticated Key Exchange (PAKE) procedure has become the most widely used protocol for secure communication on the insecure channel with the aim of making them secure. The first PAKE protocol was proposed by Bellovin and Merritt and it is named as Encrypted Key Exchange (EKE) [9]. Initially, two-party PAKE protocols gain focus among the researchers and the result of which many variations of two-party PAKE protocols have been suggested by the experts [10-12]. The two-party PAKE protocols are suitable for the client-server environment. As the size of the environment kept on increasing, the need for the three-party PAKE (3PAKE) begun which is suitable for client-client

communication. The example of such conventional three-party applications are Kerberos [13] and KryptoKnight [14] however, these applications are susceptible to password guessing attacks.

The Encrypted Key Exchange (EKE) protocol was designed to secure the communication and to protect the password from several attacks. The method is implemented using RSA, Diffie-Hellman and ElGamal asymmetric cryptosystem [9]. The method was enhanced that suits for the tree parties in communication and the several weaknesses in the EKE protocol are rectified by the Steiner and Tsudik [15]. Unluckily, the method is vulnerable to off-line guessing attacks. LSH 3-party EKE was proposed that are resistant to on-line guessing and off-line guessing attacks by using the server's public key [16]. An improvement was made on the method and the authors suggested two protocols password based and verifier based SCH 3-party EKE [17]. Howevr, the method is capable of accessing private transmission.

Another method was introduced that uses one-time key generation using Diffie-Hellman key exchange instead of using server's public key [18]. CC 3-party EKE is another variation that uses symmetrical encryption and provides mutual authentication [19]. A novel and simple three-party encrypted key exchange protocol (S-3PAKE) [20] was suggested in which the method is compromised by several vulnerabilities [21]. The first Chaos-based 3PAKE Protocol was introduced that inherits several security features with high computational efficiency [22, 23]. Another variation of the system using symmetric cryptosystems and server's public key was introduced that protects the system against various attacks [24].

In [25], chaos-based 3PAKE protocols was introduced but the difference is that the method works without smart cards. Another variation with the first chaotic map based three-party password authenticated key exchange protocol without timestamps was introduced [26]. The method was suggested with three-party authenticated key agreement protocol based on chaotic maps without password and clock synchronization to prevent password-related attacks [27]. A novel method for key exchange was developed and termed as Password Authenticated Key Exchange protocol by Juggling (J-PAKE) [28]. This method does not need any PKI deployments and allows protection against password leaks. Abdalla et al. implemented most efficient SPOKE (Simple Password-only Key Exchange) scheme in which improvement has been made in computational complexity [29]. This method employs Smooth Projective Hash Functions (SPHF) [30]. The comparison of these methods and the survey on the existing PAKE protocols are presented along with their merits in a detail by Vollala et al.[31].

Another 3PAKE based scheme was suggested that utilizes verifiers to resist various security attacks particularly different dictionary attacks [32]. However, the method is susceptible to offline dictionary. This method was improved by introducing the verifiers. The method avoids the use of public key of the server and symmetric encryption/decryption algorithms [33]. A robust and efficient 3PAKE scheme was introduced that doesn't use the public key of server and encryption/decryption algorithms with round efficiency. However, the user can be imitated by the adversary without having the password [34]. A modest scheme was introduced that maintains the computation costs and offers security to resist popular security attacks [35]. Apart from the various PAKE protocol, there are other methods that were suggested to protect the web and its user from several vulnerabilities. A new protocol named Direct Validation of Certificates (DVCert) was introduced [36] that directly validate the certificates using existing authentication information instead of relying on third-parties for certificate validation [37].

Several security indicators were suggested by the research community for attack identification. Unfortunately, due to high false rate produced by these indicators, normal web users possibly pay no attention towards using these security indicators. Also, due to lack of training, even more effective security indicators that were proposed are not extensively accepted and implemented by the majority of web clients and retailers [38]. As with the growing technology, several web browsers employ built-in "anti-phishing" filters with special functionalities. These filters will not protect users against phishing attacks completely, since, approximately, 10% of the phishing sites are still not strained by these built-in functionalities [39].

To protect the client from the password stealing attacks, a Digest Web Authentication procedure using a hash function to communicate the challenge and response between the client and server was introduced. The idea behind this method is to hash the values without sending the plain text in the communication medium. The MD5 hash function is used to hash the values. Unfortunately, it does not protect the user from forwarding attacks, since, the phisher can forward the hash values to the legitimate site to succeed the authentication, thereby collects the intended information from the server [40]. The Mutual Authentication Protocol for HTTP also drops the idea of trusting third parties and combines user authentication with SSL/TLS channel binding. As an alternative for using certificates, it employs a user's password. However, this mechanism involves supplementary server state [41].

IV. SIGNATURE BASED KEY EXCHANGE (SKE)

The proposed method provides secure communication between the web user and the web application. It prevents the data from data stealing attacks and phishing attacks. The base of the work is considered as a Diffie-Hellman Key Exchange for mutual authentication between client and server. The main focus of the work is to provide the three main security policies such as confidentiality, integrity and authenticity. The method further employs the shared secret key to access sensitive information and maintains confidentiality and integrity.

As in Diffie-Hellman Key Exchange, the two parties share a prime p and a base g. Along with the user name and password, the domain URL which is considered as a fingerprint or signature of any application is used to exchange

the keys between the two parties. Obviously, the domain URL is unique for an application and the adversary will have a slightly different URL than the legitimate web applications, though the visual appearances are similar [42]. Apart from prime and base, the two parties will have a common hash function and encryption algorithm. The three phase mechanism of the proposed method is explained in this section.

A. Registration Phase

The registration phase is important for any application as the user details are stored in the server's application for later user. It is normally a procedure to become a member of the web application. The user provides general details along with the username (U_r), password (P_r) with the web application. However, the password is to be chosen in such a way that it must include the characters, numbers and symbols to avoid password guessing attacks. Instead of sending the data directly, the hashed values are transmitted over the communication medium through the client. The details are {U_r, $H(P_r)$, $H(D_r)$, $H(D_{url})$}. $H(D_r)$ represents the other user details that are hashed. In this paper, the client and the user are used interchangeably in which the client represents the web browser and the user represents the humans operating the web browsers. The detailed picture of the registration phase is shown in Fig.2.

On receiving the details, the web application calculates the salt value S_v which is considered as the signature of the user by hashing their username, password and the domain URL (D_{url}) of the application as in (1). The terms, web application and server are used interchangeably in which the web application is an application through which the clients are serviced and the server is a web server where the web applications are hosted. All the client details along with the salt value are stored in its user database except password.

$$S_v = H(U_r, P_r, D_{url}) \quad (1)$$

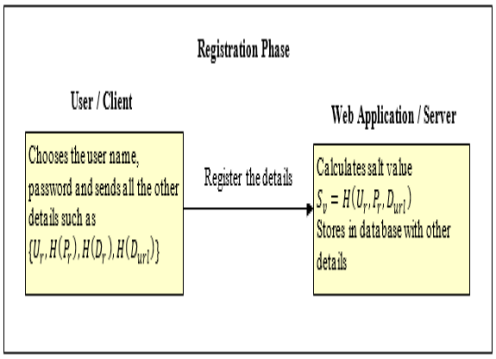


Fig. 2 Registration Phase of the Proposed Model

B. Authentication Phase

Generally, once the user becomes a member of the application, each time when the user requests the service from the web application, the web application responds with the login page. Once the login is successful, the client and web application communicate for the required services. Thus, in the proposed method, the client sends the request and the web application responds with the login page along with the shared public values prime p and base g. The client (web browser) on acquiring the application's home page, allows the user to enter the username U_r and password P_r. Thus the client calculates the salt value S_v using username, password and domainURL as shown in (2).

$$S_v = H(U_r, P_r, D_{url}) \quad (2)$$

On calculating the salt value, the client sends the username U_r, hashed password $H(P_r)$, hashed domainURL $H(D_{url})$ and the calculated salt value (S_v) to the application. On receiving the values, the web application verifies it with the database. If the authentication holds, ie., the calculated S_v matches with the one in the database, the server calculates its own cryptographic value V_s using the salt value S_v, a random R_s, prime p and base g as in (3) and forwards the value V_s to the client as a successful authentication

$$V_s = (g^{S_v})^{R_s} \bmod p \quad (3)$$

Thus, the password will not be transmitted directly over the communication medium and it is unknown to the attacker even it is hacked. This step prevents the password stealing attacks. On receiving the cryptographic value V_s, the client computes its own cryptographic value V_c using the salt value S_v, a random R_c, prime p and base g as given in (4).

$$V_c = (g^{S_v})^{R_c} \bmod p \quad (4)$$

On receiving the client's cryptographic value V_c, the server computes the shared secret key K_{sk} as shown in (5) and the client computes the shared secret key using the received server's cryptographic values V_s as shown in (6).

$$K_{sk} = (V_c)^{R_s} \bmod p \quad (5)$$

$$K_{sk} = (V_s)^{R_c} \bmod p \quad (6)$$

Once the shared secret key is computed both by the client and the server, the next step is the mutual authentication of client and server. Thus, both the client and server calculates its own authentication key. The client computes the client authentic key C_{ak} using the shared secret key, username, salt value, domainURL, cryptographic values of client and server as shown in (7) and sends it to the server.

$$C_{ak} = H(K_{sk}, U_r, S_v, D_{url}, V_s, V_c) \quad (7)$$

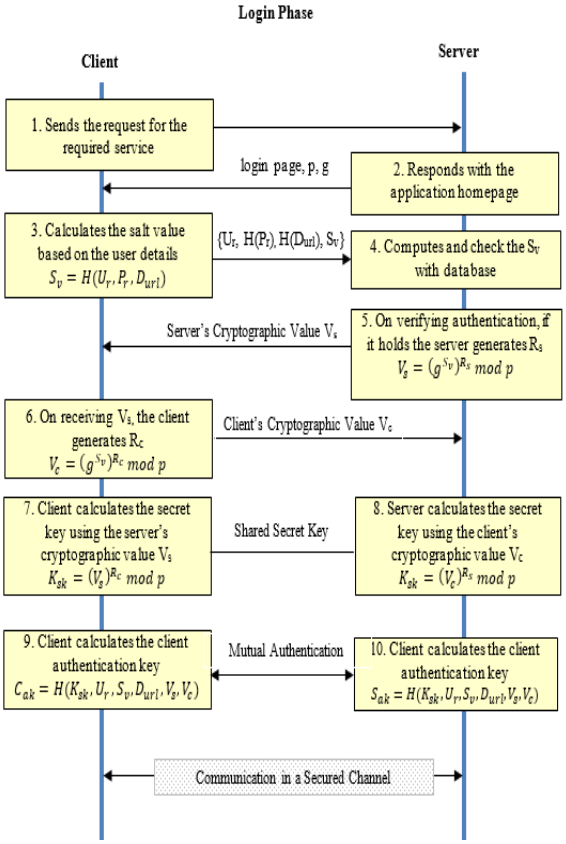
Similarly, the server calculates the server authentication key S_{ak} using the same parameters used by the client as given in (8) and sends it to the client.

$$S_{ak} = H(K_{sk}, U_r, S_v, D_{url}, V_s, V_c) \quad (8)$$

If the authentication keys of client and server are same then, the authentication is succeeded. Then, the user and web application can communicate with their sensitive data.

Thus, this mutual authentication step prevents the application and user data from the credential forwarding attacks and MITM attacks. Further, if the client tries to access any sensitive value from the application database, the values are transmitted over the secure channel with encryption using a computed shared secret key. The entire communication takes place through the secure channel. The client and server can decrypt using the shared secret key. The sequence diagram for the proposed framework is depicted in Fig. 3.

Fig. 3 Authentication Phase of the Proposed Model



C. Password Update Phase

Another important phase of the proposed method is the update password. If the user wishes to change the password, the user sends the request to the application through the client. On seeing the request, the server sends the webpage for changing the password. The user enters the old password and new password along with the username. The client computes the hash values for the old password (OP_r) and new password (NP_r). The details $\{U_r, H(OP_r), H(NP_r), H(D_{url})\}$ is sent to the web application. On receiving the details, the web application verifies the old password by computing the salt value. If it matches with the stored salt value, then it computes the salt value for the new password and then updates the new salt value in its user database. The sequence diagram for the password update phase is shown in Fig. 4.

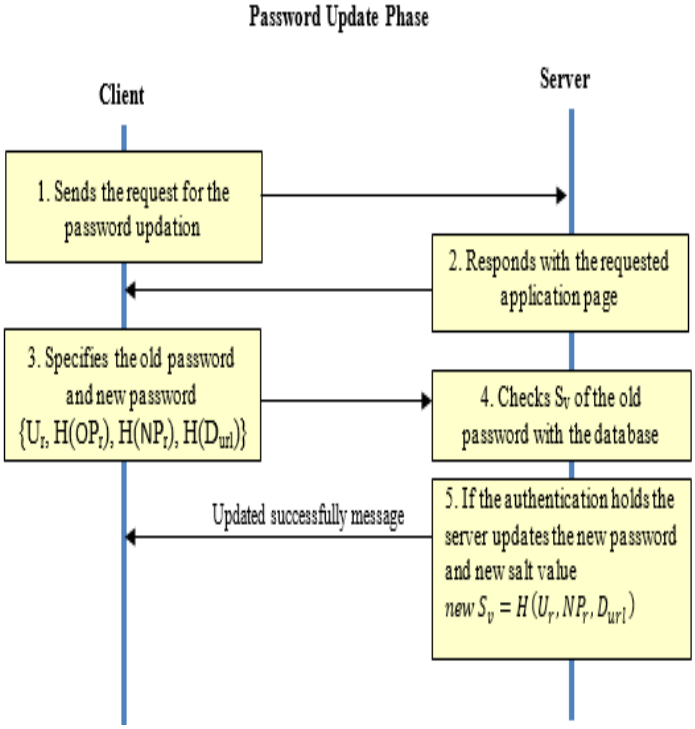


Fig. 4 Password Update Phase of the Proposed Model

The main advantage of the proposed signature based key exchange is that the password is communicated in a secure way. Also, the secret key is never transmitted directly over the communication medium. This protocol requires a little modification on the client and server side. As in Diffie Hellman key exchange, the selection of p and g are arbitrary. Thus, the proposed method to be secure, p can be chosen with 1024 bits and g with 308 bits. The universally accepted hash algorithm is used in this proposed method which is the most popular MD5 algorithm with a digest of 128 bits. For encryption and for sending the message in a secured communication channel, AES algorithm is employed along with the computed shared secret key.

V. SECURITY ANALYSIS

The proposed method is analyzed based on various policies and resistance against various data stealing attacks. Also, the proposed method is compared with other existing techniques.

A. Analyses on various security policies

The security analysis of the proposed method and the security policies against various vulnerabilities are explained in this section.

1) Mutual authentication: In case of preventing data theft and data stealing attacks, both the client and server must authenticate each other. This two-way authentication is termed as mutual authentication which is significant for preventing the masquerade and imitation attacks. In the proposed signature based method, both the client and server mutually authenticate using the authentication key and for further communication.

2) Use of Password Table: Generally, the passwords are stored directly in the user password table on the server system. This is the most attractive place and is focused by the adversary for fetching the credentials about the user. If this happens, then all the users registered in the web application will become vulnerable. Thus the proposed method does not use the password table. The password is not directly stored on the server system but the salt value computed from the user name, password and the domain URL is stored. Whenever the user logged into the system, the salt value is computed and if it is matched with the stored one, the application will allow the user to the next level authentication.

3) Password Guessing Attacks: First and foremost attack that is performed by the 50% of the human is the password guessing attack. The password of the legitimate user is predicted by random and anonymous guess. The other significant variations are dictionary attacks and brute force attacks. Using dictionary attack the adversary tries hundreds of common words as in a dictionary. The brute force attack is the process of trying all possible words and phrases as a guess for the password until the correct one is found. Thus, the password guessing attacks can be prevented by insisting the user to choose the password with alphabets, numbers and symbols or special characters. Also, while logging into the system the time interval for entering the username and password and providing only the minimum number of chances to enter the password will eventually prevent the user and application from password guessing attacks.

4) Replay attack: In this type of attack, the data transmitted over the communication channel is fraudulently retransmitted. Here, the adversary steals the sensitive information such as shared key during the legitimate communication between the legitimate clients and reuses as a masquerade activity. However, in the proposed method the random numbers which are created only once, indicating the freshness of the message is used by the communication parties to prevent the replay attack.

5) Modification Attacks: In this type of attack, the attacker tries to interpret the message and modifies the data which is very difficult to detect and prevent. As the data are transmitted in a secure way using the secret key and the authentication key, the proposed signature based key exchange method prevents the modification attacks.

6) Stolen Verifier Attacks: The attacker steals the verification data from the server in the current or past authentication sessions. The adversary generates the communication data based on the stolen data and sends them to the server as a legitimate user. This will be prevented by eliminating the storage of password on the server system.

7) Man-in-the-Middle Attacks: In this attack, the intruder listens to the communication between the two legitimate parties. Either the adversary interprets the sensitive data or in some cases he modifies the data by attempting a modification attack. This type of attack is the base for data stealing.

8) Impersonation Attacks: With this attack, the attacker tries to imitate the legitimate user or as a legitimate application. This is similar to the man in the middle and modification attacks. Using the proposed method, this attack gets prevented using the domain URL as one of the parameters. Additionally, the adversary needs to compute the authentication key and secret key using the prime p and base g along with the random numbers.

9) User Anonymity: This security policy hides the real details and identity about the user to the outside world as the adversary interprets the secret information about the user during the legitimate communication. In the proposed method, the details are not directly transmitted over the communication channel. Instead, the hashed values are transmitted.

Table 1 compares the various authentication schemes that are discussed in the literature with respect to the security requirements. The existing method such as STW 3-Party EKE Protocol [15], LSH 3-Party EKE Protocol [16], SCH 3-Party EKE Protocol [17], LSSH 3-Party EKE Protocol [18], CC 3-Party EKE Protocol [19], LC 3-Party EKE Protocol [20], LXLY [22], CLC [23], ZGL [24], LLH [25], XYZ [26], LNKK [27], JPAKE [28], SPOKE [29], VPAKE [33], 3PAKE [34], M3PAKE [35] are compared against the proposed signature based key exchange.

In order to analyse the performance of the proposed model, an experiment has been made with a simple client and server setup. A server machine has been installed with database storage using MySQL. Several requests have been sent from the client system to analyse the CPU utilization. The CPU Utilization on the server and client are computed and the details are depicted as a graph. The CPU utilization on the server system and the client system are shown in Fig. 5 and Fig. 6 respectively.

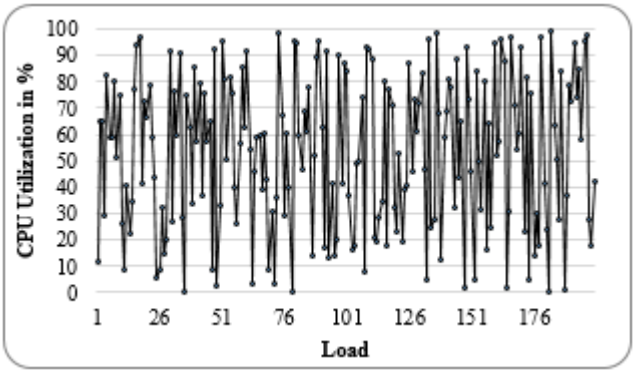


Fig. 5. CPU Utilization on Server System

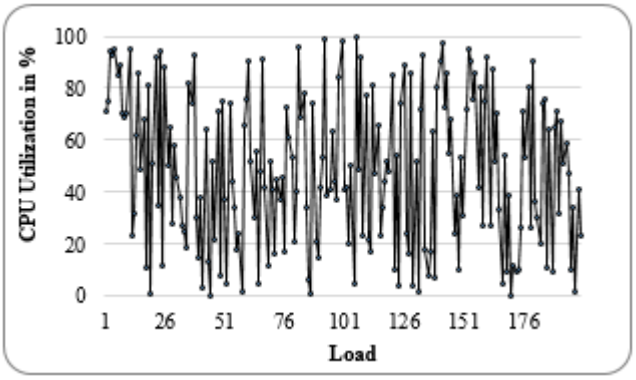


Fig. 6. CPU Utilization on Client System

TABLE I. COMPARISON OF VARIOUS SECURITY REQUIREMENTS

Security policies and resistance to attacks	STW	LSH	SCH	LSSH	CC	LC	LXLY	CLC	ZGL	LLH	XYZ	LNKK	JPAKE	SPOKE	VPAKE	3PAKE	M3PAKE	SKE
Mutual authentication	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Use of Password Table	Y	Y	Y	Y	Y	N	Y	N	Y	N	Y	N	N	N	N	N	N	N
Resist to Password Guessing Attack	N	N	N	N	N	N	N	N	Y	Y	N	N/A	Y	Y	Y	Y	Y	Y
Resist to Replay attack	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y
Resist to Modification Attack	N	N	N	N	N	N	Y	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y
Resist to Stolen Verifier Attack	N	N	N	N	N	Y	Y	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y
Resist to Man-in-the-Middle Attack	N	N	N	N	N	N	Y	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y
Resist to Impersonation Attack	N	N	N	N	N	Y	Y	N	Y	Y	Y	Y	N	N	N	N	Y	Y
User Anonymity	N	N	N	N	N	N	Y	N	Y	Y	Y	Y	N	N	Y	Y	N	Y

Upon implementing the proposed signature based key exchange, the execution time has been measured. To measure the time overhead of the proposed method, the time delay has been calculated by comparing the execution time with and without the proposed framework. The response time is calculated on the client side to mutually authenticate the application and the time delay is computed with the result as shown in Fig 7. However, the average delay in the response time is approximately 4 seconds which is meagre when compared to the security vulnerabilities.

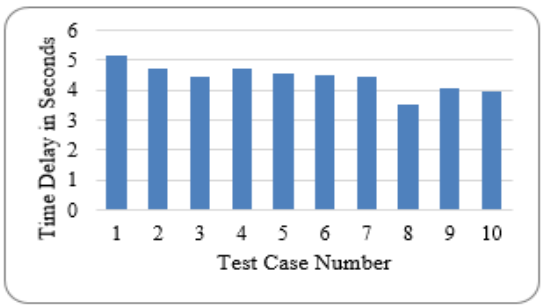


Fig. 7. Response Time Delay for various test cases

B. Comparison of SKE with other Techniques

1) SKE versus Digital Certificates: Generally digital certificate duly signed by the third party certificate authority is used for authentication between two communication parties to avoid various data stealing and eavesdropping [36]. It is based on public and private key pairs with Public Key Infrastructure (PKI) environment. However, in recent days this PKI environment which is very important in creating and maintaining this digital certificate is becoming weak. Several ways such as Stolen code-signing certificates, Malware installed illegitimate certificates, MITM attacks abused certificates and weak or improper certificates, the fake or weak certificates are issued to sign the malicious software which has been misused by the attacks on private and public organizations. Also, the CA's are conceded for stealing the

data by implementing MITM attacks. In the proposed SKE Algorithm, third party attestation is not used as it cannot be trusted. However, the two communicating parties use self-attestation and mutual authentication. Also, SKE algorithm is resistant to MITM attacks.

2) SKE versus Security Indicators: Since it is very difficult to create a black list for phishing sites which increase in number day by day, security indicators are used to ensure the trustworthy of the site [38]. The main disadvantage is the normal web user may not look for them since they don't have much knowledge about those indicators in a web browser. Additionally, the indicators can mislead the user to the hijacked site through web vulnerabilities such as cross-site scripting. In SKE Algorithm, the domain URL of the application is used at the first step as an anti-phishing mechanism, since the URL is a fingerprint of any application. The main advantage is that user will not need any prior knowledge or training for security concerns.

3) SKE versus Anti-Phishing Filters: Several anti-phishing filters have been used with special functionalities to avoid phishing attacks [39]. However, it does not protect completely since only 90% of the phishing sites have been identified by these built-in functionalities. Also, the details about the new phishing sites have to be updated regularly. Phishers sometimes bypass the filter's database by slightly altering the words. In SKE algorithm, due to mutual authentication and authorization, the user will authenticate to the legitimate site by using the salt value. For each time, when the user login to the site, the salt value is calculated. Only the legitimate user can login to the legitimate web application.

4) SKE versus Digest Web Authentication: In Digest Web Authentication, the hash values are used instead of sending the plain text in the communication medium [40]. Unfortunately, it suffers from forwarding attacks, since, the phisher can forward the hash values to the legitimate site to succeed the authentication, thereby collects the intended information from the server. In SKE algorithm, even the attacker forwards the credentials, he cannot receive the data since the authentication step has to be completed to calculate shared secret key which needs salt value, username and password. However, this step will not be completed by the attacker since the password is not communicated directly through the channel at any point of time.

VI. CONCLUSION

In this paper, a signature based key exchange approach has been presented that allows improving the integrity and confidentiality of a client and the server through mutual attestation and authentication over protected data. Instead of trusting the third party for integrity and security, the two communicating systems can authenticate and attest themselves mutually through the generation of the authentication key. This system relies on a key exchange based on the password, username and the corresponding salt value. Various security policies and resistant to various attacks are presented and are compared with the existing techniques. Thus, the SKE method prevents the user and web application from data theft. The future work concentrates on evaluating the system with different real world application to protect themselves from data theft. Also, the proposed method has to be extended to support the three parties for client to client communication and with multiserver communication.

REFERENCES

1. The Open Web Application Security Project (OWASP). https://www.owasp.org.
2. OWASP, Top 10 Application Security Risks - 2017. Open Web Application Security Project (OWASP).
3. Setu Kulkarni, 2018 WhiteHat Application Security Statistics Report is a Call to Arms for DevOps Teams, October 04, 2018.
4. Micro Focus, Fortify Software Security Research Team, 2018 Application Security Research Update, 2018
5. A.K.Jain, and B.B. Gupta, "Phishing detection: analysis of visual similarity based approaches", Security and Communication Networks, 2017.
6. R. Savita, and U. Datta, "Two Way Authentication in MITM Attack to Enhance Security of E-commerce Transactions", International Journal of Security and Its Applications, 9(9), pp.265-274, 2015
7. Pfleeger and Pfleeger, Security in computing, Prentice Hall Professional Technical Reference, 2007.
8. S. Son, Toward better server-side Web security, Diss., The University of Texas at Austin, 2014.
9. Steven M Bellovin and Michael Merritt, "Encrypted key exchange: Password based protocols secure against dictionary attacks", In Proc. of the symposium on Research in Security and Privacy, pages 72–84. IEEE, 1992.
10. E.Bresson, O.Chevassut, D.Pointcheval, "New security results on encrypted key exchange", In: Proc PKC, LNCS 2947, pp 145–158, 2004.
11. M. Abdalla, D. Pointcheval, "Simple password-based encrypted key exchange protocols", In: Proc of topics in cryptology, CT-RSA 2005, LNCS 3376, pp 191–208.
12. M.Abdalla, O.Chevassut, D.Pointcheval, "One-time verifier-based encrypted key exchange", In: Proc of PKC '05, LNCS 3386, pp 47–64, 2005
13. B.Clifford Neuman and Theodore Ts'o. Kerberos: An authentication service for computer networks, IEEE Communications magazine, 32(9):33–38, 1994.
14. Refik Molva, Gene Tsudik, Els Van Herreweghen, and Stefano Zatti, "Kryptoknight authentication and key distribution system", In European Symposium on Research in Computer Security, pages 155–174. Springer, 1992.
15. Michael Steiner, Gene Tsudik, and Michael Waidner, "Refinement and extension of encrypted key exchange", ACM SIGOPS Operating Systems Review, 29(3):22–30, 1995.
16. Chun-Li Lin, Hung-Min Sun, and Tzonelih Hwang, "Three-party encrypted key exchange: attacks and a solution", ACM SIGOPS Operating Systems Review, 34(4):12–20, 2000.
17. Hung-Min Sun, Bing-Chang Chen, and Tzonelih Hwang, "Secure key agreement protocols for three-party against guessing attacks", Journal of Systems and Software, 75(1):63–68, 2005.
18. Chun-Li Lin, Hung-Min Sun, Michael Steiner, and Tzonelih Hwang, "Three-party encrypted key exchange without server public-keys", IEEE Communications letters, 5(12):497–499, 2001.
19. Chin-Chen Chang and Ya-Fen Chang, "A novel three-party encrypted key exchange protocol", Computer Standards & Interfaces, 26(5):471–476, 2004.

20. Rongxing Lu and Zhenfu Cao, "Simple three-party key exchange protocol", Computers & Security, 26(1):94–97, 2007.
21. Raphael C-W Phan, Wei-Chuen Yau, and Bok-Min Goi, "Cryptanalysis of simple three-party key exchange protocol (s-3pake)", Information sciences, 178(13):2849–2856, 2008.
22. Hong Lai, Jinghua Xiao, Lixiang Li, and Yixian Yang, "Applying semigroup property of enhanced chebyshev polynomials to anonymous authentication protocol", Mathematical Problems in Engineering, 2012.
23. Tzung-Her Chen, Wei-Bin Lee, and Hsing-Bai Chen, "A round- and computation efficient three-party authenticated key exchange protocol", Journal of Systems and Software, 81(9):1581–1590, 2008.
24. Fengjun Zhao, Peng Gong, Shuai Li, Mingguan Li, and Ping Li, "Cryptanalysis and improvement of a three-party key agreement protocol using enhanced chebyshev polynomials", Nonlinear Dynamics, 74(1-2):419–427, 2013.
25. Cheng-Chi Lee, Chun-Ta Li, and Che-Wei Hsu, "A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps", Nonlinear Dynamics, 73(1-2):125–132, 2013.
26. Qi Xie, Jianmin Zhao, and Xiuyuan Yu, "Chaotic maps-based three-party password-authenticated key agreement scheme", Nonlinear Dynamics, 74(4):1021–1027, 2013.
27. Xiong Li, Jianwei Niu, Saru Kumari, Muhammad Khurram Khan, Junguo Liao, and Wei Liang, "Design and analysis of a chaotic maps-based three-party authenticated key agreement protocol", Nonlinear Dynamics, 80(3):1209–1220, 2015.
28. Feng Hao and Peter Ryan, "J-pake: authenticated key exchange without PKI", In Transactions on computational science XI, pages 192–206. Springer, 2010.
29. M. Abdalla, F. Benhamouda, D. Pointcheval, "Spoke: Simple password-only key exchange in the standard model", IACR Cryptology ePrint Archive, 2014:609, 2014.
30. R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption", In International Conference on the Theory and Applications of Cryptographic Techniques, pages 45–64. Springer, 2002.
31. S. Vollala, B.S. Begum, and N.Ramasubramanian, "Evaluation of password encrypted key exchange authentication techniques: design approach perspective: evaluation of PAKE protocol", In Proceedings of the International Conference on Internet of Things and Machine Learning (p. 16). ACM, 2017.
32. H.Chien, "Secure verifier-based three-party key exchange in the random oracle model", Journal of Information Science and Engineering, 27(4), 1487–1501, 2011.
33. Q.Pu, J.Wang, S.Wu, & J.Fu, "Secure verifier-based three-party password-authenticated key exchange", Peer-to-Peer Networking and Applications, 6(1), 15–25, 2013.
34. T.Y.Youn, E.S.Kang, & C.Lee, "Efficient three-party key exchange protocols with round efficiency", Telecommunication Systems, 52(2), 1367–1376, 2013.
35. M.Heydari, S.M.S.Sadough, M.S.Farash, S.A.Chaudhry, and K.Mahmood, "An efficient password-based authenticated key exchange protocol with provable security for mobile client–client networks", Wireless Personal Communications, 88(2), pp.337-356, 2016.
36. X.Li, S.Li, J.Hao, Z.Feng, and B.An, "Optimal Personalized Defense Strategy Against Man-In-The-Middle Attack", In AAAI (pp. 593-599), 2017.
37. M.Cui, Z.Cao, and G.Xiong, "How Is the Forged Certificates in the Wild: Practice on Large-Scale SSL Usage Measurement and Analysis", In International Conference on Computational Science (pp. 654-667). Springer, Cham, 2018.
38. C.Marforio, R.Jayaram Masti, C.Soriente, K.Kostiainen, and S.Čapkun, "Evaluation of personalized security indicators as an anti-phishing mechanism for smartphone applications", In Proceedings of the CHI Conference on Human Factors in Computing Systems (pp. 540-551). ACM, 2016.
39. A.Laszka, J.Lou, and Y.Vorobeychik, "Multidefender strategic filtering against spear-phishing attacks", In Thirtieth AAAI Conference on Artificial Intelligence, 2016.
40. M.Manulis, D.Stebila, F.Kiefer, and N.Denham, "Secure modular password authentication for the web using channel bindings", International Journal of Information Security, 15(6), pp.597-620, 2016.
41. S.Jarecki, H.Krawczyk, and J.Xu, "OPAQUE: An Asymmetric PAKE Protocol Secure Against Pre-computation Attacks", In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 456-486). Springer, Cham, 2018.
42. A.Saravanan,M.S.Irfan Ahmed, S.Sathya Bama, "Twin Shield: A Prevention Mechanism to Protect Web Data and Web User from Data Theft", International Journal of Printing, Packaging and Allied Sciences, Vol. 4, No. 3, December 2016.