Robust Authentication Scheme based on Elliptic Curve Cryptography for Big Data Applications

Geeta Sharma
Department of Computer Science and Engineering
Guru Nanak Dev University, Regional Campus
Jalandhar, Punjab, India

Sheetal Kalra
Department of Computer Science and Engineering
Guru Nanak Dev University, Regional Campus
Jalandhar, Punjab, India

ABSTRACT
In this era of big data, an unprecedented amount of data is being produced, collected and stored at an alarming rate. Apart from other issues, privacy and cyber-security have become serious as an overwhelming amount of data is collected about users both knowingly and unknowingly. The design of secure remote user authentication and key agreement schemes for big data applications is still open and challenging. Therefore, this paper proposes a robust smart card based mutual authentication and key agreement scheme for big data environment which not only resolves the security weaknesses of schemes proposed till date but also extends the scheme to provide user anonymity which is one of the most desired features of big data systems. Furthermore, the proposed scheme requires lower computation cost and provides more security and functionality features than the existing schemes. The proposed scheme thus provides more efficiency and security from any type of system intrusion.

Keywords
Big Data, Elliptic Curve Cryptography, Remote user authentication, Session Key, Smart Card, User Anonymity

1. INTRODUCTION
We live in an era of data deluge where an unprecedented amount of data is being produced, collected and stored at an alarming rate. The storage and management of this enormous data is beyond the capability of traditional tools and methods. The concept of Big Data has therefore attracted the interest of researchers for its promising ability to analyze and manage this overwhelming data efficiently. Big data refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage and analyze. Big data analytics provide promising solutions to the management of this massive data. Big data has the potential to boost efficiency, better predictions, save money and strengthen decision; thus making its applicability in a variety of fields like disaster prevention, medical studies, weather forecasting, cosmological problems that involve massive data sets, traffic control, finance, prevent fraudulent activities, business transactions, and so on. As the applicability of big data is increasing, so are the risks and threats to its applications. Apart from the challenges posed by increasing volume, velocity, variety, veracity and value, privacy and cyber-security have become serious as an overwhelming amount of data is collected about users knowingly and unknowingly. With the expeditious development of internet and wireless communication, users can easily use the services of remote server anytime and anywhere. The widespread use of cloud computing has worsen the situation as it collects much more data. The popularity of such services has also exposed the information over network to various security threats. Thus the need of practically secure user authentication and key agreement systems has become vital for such networks. Various schemes based on password, biometric, smart card, dynamic id or a combination of these have been proposed for remote user authentication so far. Of these, password and schemes employing smart card have gained more popularity because of their simplicity, scalability, efficiency and convenience. The concept of authentication based on password was introduced by Lamport [1] in 1981. He proposed a scheme based on the hash function to verify the client and the server. However, it was resistant to eavesdropping and impersonation attack, but was insecure to replay threats, offline password guessing attacks and password related problems. A number of improved password authentication schemes have been proposed since then. As more and more users register to the remote servers, their registration information is stored in a security sensitive verification table which is prone to various security attacks like insider, password guessing, server spoofing, etc. Such a table is difficult to maintain and poses a burden on the server as the number of users increase. In such situations, schemes based on smart card are favorable to provide security, efficiency and privacy as there is no requirement to store verifier table.

The paper is arranged as: Related work in the field of authentication schemes is discussed in section 2. Section 3 presents robust smart card based password authentication scheme on ECC. Section 4 presents the security analysis of the proposed scheme. Section 5 gives the functionality and performance analysis of the proposed scheme. Section 6 concludes the paper.

2. RELATED WORK
ECC based protocols gained popularity and are the strongest public-key cryptographic systems known today. Compared with RSA, Rabin and Elgamal cryptographic systems, ECC has remarkable strength and efficiency advantages in terms of bandwidth, key sizes and computational overheads. They also eliminate the problem of key distribution and digital signatures as with traditional symmetric key cryptosystems. Thus ECC when used in password authentication and update schemes provide high security at a reasonable computational cost. Unfortunately most ECC based schemes have various flaws and thus more improvement is required. In 2011, Islam and Biswas [2] studied the flaws of Lin and Hwang [3] and found it susceptible to malevolent user attack, stolen verifier, impersonation attack, many logged-in users attack, known session specific temporary information attack and proposed a secure scheme based on ECC. Their scheme also generated a common ECC based secret key which is employed for symmetric encryption. In the same year, He [4] analyzed Islam and Biswas scheme [2] and found that it is vulnerable to three kinds of attacks in different scenarios: (1) Stolen-verifier attack (2) Offline password guessing attack (3) Privileged insider attack. Further, Wang, Juang and Lei [5] in 2011 studied Wang et al. [6] scheme and found it vulnerable to lost smart card problem and known key attack. They further proposed a scheme based on the elliptic curve discrete logarithm problem. In 2012, He, Wu and Chen [7] performed a cryptanalysis of Islam and Biswas scheme [2] and found that scheme insecure to offline password guessing attack and stolen-verifier attack. In the same year Wang et al. [8] also analyzed Islam and Biswas scheme [2]

and revealed following weaknesses: (1) It is susceptible to offline password guessing attack, stolen verifier attack and denial of service (DoS) attack; (2) It is unable to maintain user anonymity. Further, C. T. Li [9] also analyzed Islam and Biswas scheme [2] and it is prone to offline password guessing attack, stolen-verifier and insider attacks. He further presented a smart card based ECC scheme that also provides user anonymity. In 2014, Wang [10] demonstrated that in addition to previously found security flaws [4, 8, 9] in Islam and Biswas scheme [2] like offline password guessing attack, stolen verifier attack, privilege insider attack, and denial of service attack, their scheme cannot resist password compromise impersonation attack. She further proposed an anonymous remote authentication scheme using smart card without using bilinear pairing computation. She claimed that her scheme not only inherits the advantages in Islam and Biswas' scheme, but also provides more features, including preserving user anonymity, supporting offline password change, revocation, re-registration with the same identifier and system update. Also, Qiao and Tu [11] proposed a security enhanced scheme that eliminates the weaknesses of Islam and Biswas scheme [2] as pointed out by He et al. [7]. They claimed that their scheme performs better than [2]. Ramesh and Bhaskaran [12] in 2014 analyzed Li scheme [9] and demonstrated that it is prone to insider attack, password guessing attack, stolen verifier attack and fails to ensure user anonymity. It is also inefficient in error password login. They further proposed an improved scheme which inherits merits of [9] with the removal of modular computations involved in bilinear pairing operations. Song [14] proposed a smart card based password authentication protocol. In 2015, Kalra and Sood [15] proposed a secure scheme to authenticate cloud servers and IoT using Elliptic Curve Cryptography (ECC). In 2016, Sharma and Kalra [16, 17] proposed authentication schemes employing quantum identity to authenticate a user and the cloud server.

Table 1. Notations used in the proposed scheme

Notations used	Description
ID_A	Identity of the client *A*
pw_A	Secret password of the client *A*
d_S	Secret key of the server *S*
$h_1( )$	One-way hash known only to client *A*
$h_2( )$	One-way hash known only to server *S*
$h( )$	One-way hash known to both *A* and *S*
U_A	Password verifier of client *A*, where $U_A = h_1(pw_A).G$
U_S	Public key of the server *S*, where $U_S = h_2(d_S).G$
K_x	Secret key computed either using $K = h_1(pw_A).U_S = (K_x, K_y)$ or $K = U_A . h_2(d_S) = (K_x, K_y)$
$E_{Kx}( ) / D_{Kx}$	Symmetric AES with key K_x
G	Bases point of the elliptic curve group of order n such that $n.G = O$
r_A / r_S	Random numbers
||	Concatenation operator of two strings
$\oplus$	Exclusive OR (XOR) operation
$\rightarrow$	Public channel
$\Rightarrow$	Secure channel

3. PROPOSED SCHEME

The proposed scheme consists of following six phases: Registration phase, Pre-computation phase, Password authentication and key agreement phase, Password change phase, User eviction phase and User anonymity phase. Table 1 denotes the parameters used in the proposed scheme. Fig.1. depicts the proposed scheme.

3.1 Registration Phase

The server *S* chooses a large prime number *p* and integers *a*, *b*, where $4a^3+27b^2 \neq 0$. *S* then selects an elliptic curve equation E_p over the finite field *p* represented as: $y^2 mod\ p \equiv x^3+ax+b\ (mod\ p)$. Let $G=(x_1, y_1)$ be a base point in $E_p(a,b)$ whose order is an extremely large value *n*. The order *n* of a point *G* on an elliptic curve is the smallest positive integer *n* such that $n.G=O$ where *O* is a point of the elliptic curve at infinity.

Step 1. The client *A* registers with server *S* with his/her own ID_A and password verifier U_A where $U_A = h_1(pw_A).G$ and sends (ID_A, U_A) to server *S* over a secure channel.

***Step* 2.** After verifying the validity of *A*, *S* evaluates $U_S = h_2(d_S).G$ and $Z_A = U_S \oplus U_A$ and stores identity, Z_A and status-bit in the verifier table on server side as shown in Table 2. The status bit represents the login status where bit 1 indicates that the user is logged into the server and bit 0 indicates that it is not logged into the server. *S* then writes (ID_A, U_S, G, E_p) into smart card S_M and issues it to *A* over a safe channel.

Table 2. The verifier table

Identity	Z	Status-bit
ID_A	$Z_A = U_S \oplus U_A$	0/1
ID_B	$Z_B = U_S \oplus U_B$	0/1
ID_C	$Z_C = U_S \oplus U_C$	0/1
…	…	…

***Step* 3.** On receiving the smart card S_M from the *S*, *A* activates it by inserting into the card reader. The client then keys in password pw_A. S_M calculates a digest value of the password $h_1(pw_A), h_1(pw_A).G$ and $U_S' = U_S \oplus U_A$. S_M replaces U_S with U_S'. Finally, S_M contains following parameters (ID_A, U_S', G, E_p).

3.2 Precomputation Phase

S_M chooses a random number r_A from [1, n-1] and calculates a point $R_A = r_A.G$ over equation E_p at the beginning of the authentication phase. It stores R_A into its memory.

3.3 Authentication And Key Agreement Phase

A inserts the personal smart card S_M and inputs password pw_A. Then S_M and *S* performs steps:

Step 1. Client →Server: $(ID_A, E_{Kx}(ID_A, R_A, W_A))$

The smart card first evaluates $U_S = U_S' \oplus U_A = h_2(d_S).G$. The smart card S_M computes $W_A = r_A . U_S = r_A . h_2(d_S).G$. It then encrypts the parameters ID_A, R_A, W_A using symmetric key K_x and sends $E_{Kx}(ID_A, R_A, W_A)$ to the server *S*. The encryption key

K_x is the x coordinate of K= $h_1 (pw_A).U_S = h_1 (pw_A).h_2 (d_S) .G = (K_x, K_y)$.

***Step* 2.** Server →Client: R_S, $h (W_S || W_A)$

On receiving the login request $(ID_A, E_{Kx}(ID_A, R_A, W_A))$, *S* first verifies the authenticity of client's identity by checking the received ID_A with ID_A stored in the verifier table on the server side. Then *S* computes $U_A = Z_A \oplus U_S$ and decrypts $E_{Kx}(ID_A, R_A, W_A)$ using the decryption key K_x by calculating $K = U_A . h_2 (d_S) = h_1 (pw_A). h_2 (d_S) .G = (K_x, K_y)$ and obtains ID_A, R_A and W_A.*S* then verifies the received ID_A to the identity obtained on decrypting $E_{Kx}(ID_A, R_A, W_A)$. If verification fails, *S* rejects the login request. If they match, *S* calculates $W_A' = R_A .h_2 (d_S) = r_A . h_2 (d_S) .G = r_A . U_S$ and verifies computed W_A' with received W_A. If the values are equal, *S* selects a random number r_S from [1, n-1] and compute $R_S = r_S . G$ and $W_S = r_S . R_A$. Then *S* sends $R_S, h (W_S|| W_A)$ to the client.

***Step* 3.** Client →Server: $ID_A, h (r_A .U_S || W_S' || ID_A)$

On receiving the request $R_S, h (W_S || W_A)$, the smart card evaluates $W_S'= r_A . R_S$ and $h (r_A .U_S || W_S')$. It then verifies if the calculated $h (r_A .U_S || W_S')$ is equal to received $h (W_S || W_A)$. If both are equal, the identity of the server is authenticated. S_M submits a response $(ID_A, h (r_A .U_S || W_S' || ID_A))$ to *S*.

***Step* 4.** Server →Client: Access Granted/ Denied

S checks whether the received $h (r_A .U_S || W_S' || ID_A)$ is equivalent to the computed $h (W_A' || W_S || ID_A)$. If both are equal, *S* authenticates identity of the client. If all the above steps, *S* grants the client's login request else the request is declined. The client calculates the final session key as $SK= W_S' = r_A . R_S = r_A . r_S .G$ and *S* computes session key as $SK= W_S = r_S . R_A = r_A . r_S .G$.

3.4 Password Change Phase

If the client *A* wishes to change the password, he/she can insert the smart card S_M into the card reader and input the old password (pw_A) and new password (pw_{Anew}). S_M retrieves $U_S= U_S' \oplus h_1 (pw_A) .G$ and evaluates $U_S* = U_S \oplus h_1 (pw_{Anew}) .G$ using fresh password pw_{Anew}. S_M then updates U_S' with U_S* and includes (ID_A, U_S*, G, E_p). It also sends notification to server to update $U_A = h_1 (pw_A) .G$ to $U_{Anew} = h_1 (pw_{Anew}) .G$ and corresponding value of $Z_A = U_S \oplus U_A$ to $Z_{Anew} = U_S \oplus U_{Anew}$.

3.5 User Eviction Phase

Step 1. If A is removed by S, then S will remove A's identity ID_A and corresponding password verifier U_A and status-bit from the verifier table on S's side.

Step 2. When this evicted client A tries to access S by utilising the overdue data in the S_M, S can easily identify the user as the identity ID_A and the corresponding information is not available in the verifier table in Step 2 of the authentication and key agreement phase.

3.6 User Anonymity Phase

In this section, the proposed scheme is extended to ensure user anonymity, the most sought after feature of big data systems in which the intruder will not be able to trace the user's original identity. The remote server and client can however perform mutual authentication and generate a session key as depicted in 3.3.

3.6.1 Registration phase

A submits his identity and password verifier to remote server S as before. S issues an anonymous identity X_A and writes (X_A, U_S, G, E_p) to the smart card. Then, client computes U_S' and replaces U_S with U_S' as before. Finally, the smart card includes (X_A, U_S', G, E_p). Also the verifier table on the server side will replace ID_A with X_A.

3.6.2 Precomputation phase

The Precomputation phase is same as 3.2

3.6.3 Authentication and key agreement phase

Authentication is done by performing steps:

***Step* 1.** Client →Server: $(X_A, E_{Kx}(X_A, R_A, W_A))$

Step 2. Server →Client: $R_S, h (W_S || W_A)$

On receiving the login request $(X_A, E_{Kx}(X_A, R_A, W_A))$, *S* first verifies the authenticity of the identity by checking the received X_A with X_A stored in the verifier table on the server side. Then *S* computes $U_A = Z_A \oplus U_S$ and decrypts $E_{Kx}(X_A, R_A, W_A)$ using the decryption key K_x by calculating $K = U_A . h_2 (d_S) = h_1 (pw_A).h_2 (d_S) .G = (K_x, K_y)$ and obtains X_A, R_A and W_A. *S* then verifies if received X_A is equivalent to X_A obtained on decrypting $E_{Kx}(X_A, R_A, W_A)$. If condition is invalid, server declines the login request. If they match, *S* computes $W_A' = R_A .h_2 (d_S) = r_A . h_2 (d_S) .G = r_A . U_S$ and verifies if calculated W_A' is equivalent to received W_A. If the values are same, *S* selects a random number r_S from [1, n-1] and compute $R_S = r_S . G$ and $W_S = r_S . R_A$. Then *S* sends $R_S, h (W_S || W_A)$ to the client.

***Step* 3.** Client →Server: $X_A, h (r_A .U_S || W_S' || X_A)$

On receiving $R_S, h (W_S || W_A)$, the smart card computes $W_S'= r_A . R_S$ and $h (r_A . U_S || W_S')$. It then checks whether the calculated $h (r_A .U_S || W_S')$ matches with submitted $h (W_S || W_A)$. If both are equivalent, the identity of server *S* is authenticated. S_M submits a response $(ID_A, h (r_A .U_S || W_S' || X_A))$ to the server.

***Step* 4.** Server →Client: Access Granted/ Denied

S checks whether the received $h (r_A .U_S || W_S' || X_A)$ is equivalent to the computed $h (W_A' || W_S || X_A)$. If both matches, the identity of client is authenticated. If the above steps are satisfied, the server grants login access else the request is rejected. Client calculates session key $SK= W_S' = r_A . R_S = r_A . r_S .G$ and the server generates a session key $SK= W_S = r_S . R_A = r_A . r_S .G$.

4. SECURITY ANALYSIS

The security analysis of the proposed scheme is done in this section. The proposed scheme efficiently secures the system from any type of intrusion. The security attributes provided by the proposed scheme are as described below.

4.1 Password Guessing Attack

The password guessing attack is a serious problem in password based user authentication scheme. In the proposed scheme, if an intruder *E* intercepts the login message $(ID_A, E_{Kx}(ID_A, R_A, W_A))$, he cannot guess the password pw_A as he cannot compute the encryption key $K_x = h_1 (pw_A).U_S = h_1 (pw_A).h_2 (d_S) .G$ because he has no knowledge of U_S and $h_1( )$, $h_2( )$. Even if the intruder somehow manages to get these values, the value of K_x is still hard to solve in polynomial time due to Elliptic Curve Discrete Logarithm Problem (ECDLP).

4.2 Stolen Verifier Attack

In stolen verifier attack, an intruder *E* can steal the password verifier from the database and launch an offline guessing attack to acquire password pw_A of the user. In the proposed scheme, the remote server does not store password verifier U_A directly. In fact it stores $Z_A = U_S \oplus U_A = h_2 (d_S) .G \oplus h_1 (pw_A).G$ in verifier table. Even if an intruder manages to steal the value of

Z_A, it is unable to guess the password because it cannot compute the digest values using secure one-way hash functions which are further hard to solve due to Elliptic Curve Computational Diffie–Hellman Problem (ECCDHP).

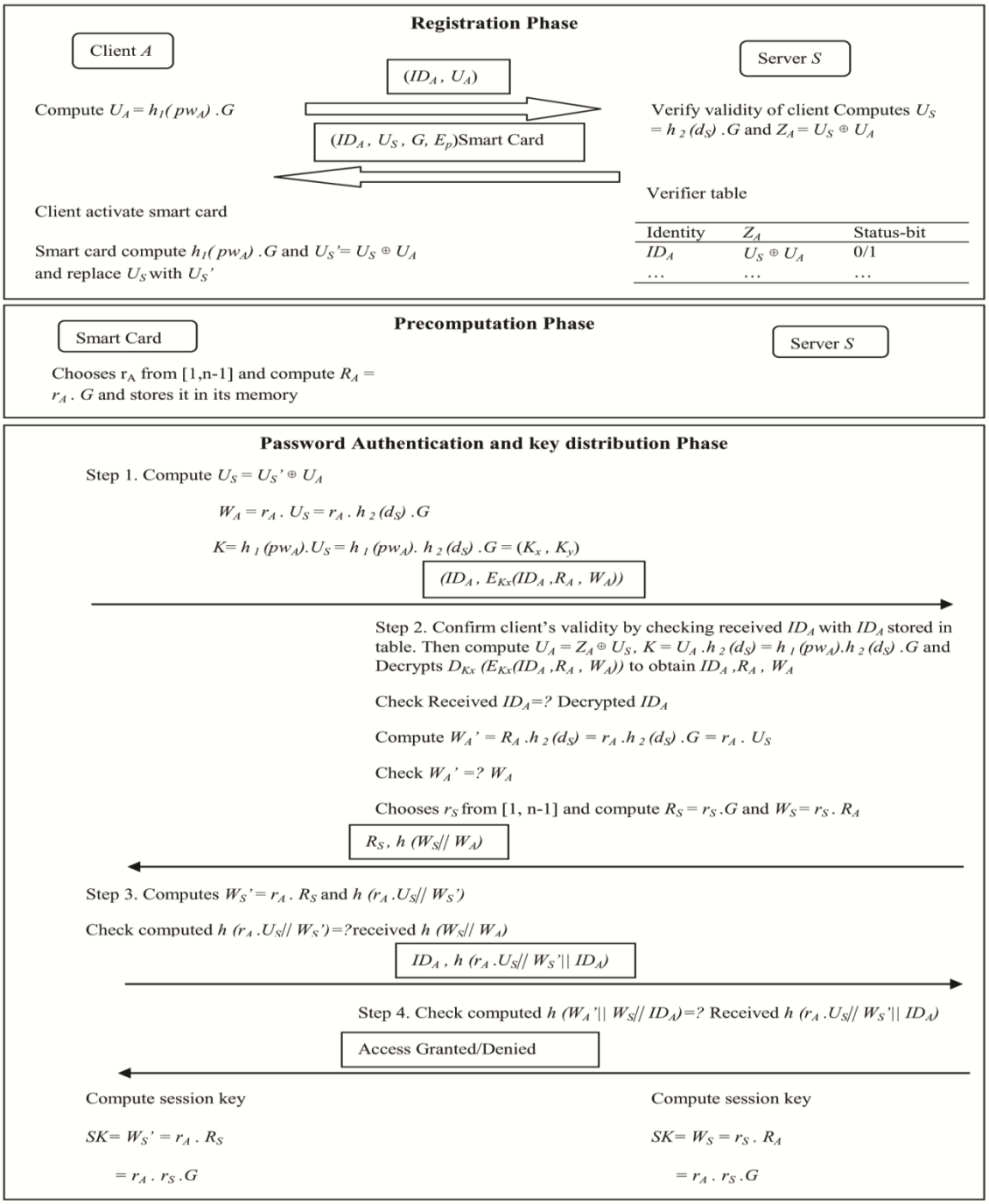

Fig.1: Proposed scheme for remote user authentication and key agreement

4.3 Insider Attack

In this attack, an insider can obtain sensitive data by pilfering verifier from the server's database. Although the proposed scheme, remote server does not store the password verifier U_A directly in the verifier table. In fact it stores $Z_A = U_S \oplus U_A$. Even if the adversary somehow manages to steal Z_A, he still cannot guess the password because of difficulties of ECDLP and ECCDHP. Thus, the scheme can resist insider attack.

4.4 Impersonation Attack

An intruder E can alter recorded message $(ID_A, E_{Kx}(ID_A, R_A, W_A))$ or imitate an authentic login message to impersonate as the original client to access the services. Although, E cannot achieve W_A as ID_A, $E_{Kx}(ID_A, R_A, W_A))$ is encrypted by a symmetric key K_x known to the client only and the server. Additionally, it is impossible to imitate an authentic login message $(ID_A, E_{Kx}(ID_A, R_A, W_A))$ as the intruder has no

knowledge of K_x and W_A, hence the intruder cannot impersonate an authentic user.

4.5 Server Spoofing Attack

In server spoofing attack, an intruder *E* can impersonate as an authentic server to get password pw_A of client and server's master key d_S. The symmetric key $K = h_1(pw_A).U_S = U_A \,.\, h_2(d_S) = h_1(pw_A).h_2(d_S).G = (K_x, K_y)$ cannot be calculated without the knowledge of values pw_A and d_S. Thus, *E* cannot imitate a legal response as in step 2 of authentication and key agreement phase as it is not feasible to obtain R_A, W_A by decrypting login request $E_{Kx}(ID_A, R_A, W_A)$ without symmetric key K_x. Thus, the proposed scheme is secure to server spoofing attack.

4.6 Many Logged-In User's Attack

The proposed scheme can resist many logged-in user's attack. It is assumed that the password (pw_A) and the identity of A (ID_A) is disclosed to a number of adversaries. In this scheme, only an intruder at a time can access the server *S* out of all who know the valid password pw_A and the identity of ID_A as *S* maintains a status-bit in the verifier table. When a client is logged in using the valid password and identity, S sets status bit = 1 and meanwhile, if any other adversary tries to access the services using the same password, S rejects all the requests because status-bit shows the user is already logged in.

4.7 Known Session-Specific Temporary Information Attack

In the proposed scheme, after mutual authentication of client *A* and server *S* conform to a session key $SK = r_A \,.\, r_S \,.G$. Suppose the password pw_A of the client and the master key of server d_S are disclosed to the intruder, it is still impossible to compute *SK* without recording random numbers $r_A$and r_S which changes with every login session. Also the computation of $r_A \,.\, r_S$ .Computing *G* is identical to solving ECDLP that is very difficult to solve by a polynomial time algorithm.

4.8 Mutual Authentication

The proposed scheme mutually authenticates client A and server S using three-way challenge-response handshake. In step 3 of authentication phase upon acquiring R_S, h ($W_S \,\|\, W_A$), the smart card computes $W_S' = r_A \,.\, R_S$ and h ($r_A \,.U_S \,\|\, W_S'$). It then verifies if calculated h ($r_A \,.U_S \,\|\, W_S'$) matches with received h ($W_S \,\|\, W_A$). If both the values are equal, the remote server is authenticated. Likewise, in step 4 of authentication phase, server checks whether the received h ($r_A \,.U_S \,\|\, W_S' \,\|\, ID_A$) is equal to the computed h ($W_A' \|\, W_S \,\|\, ID_A$). If the values are equal, the server is authenticated.

4.9 User Anonymity

During the communication between user and the remote server over an insecure network, adversary or third parties may know the identity of the client by intercepting the messages exchanged between them. Thus, providing user anonymity is very important. Therefore, the proposed scheme is extended to ensure user anonymity phase in which the identity of the user is made confidential.

4.10 Session Key Agreement

The proposed scheme generates session key SK after successful mutual authentication between the client and server in each session. The messages are communicated by encrypting data with the generated session key.

4.11 Perfect Forward Secrecy

Perfect forward secrecy ensures if long-term private keys of entities are revealed, the security of foregoing session keys initiated by legal entities is not affected. If client A's password pw_A or password verifier U_A is revealed, it does not permit the intruder to calculate $SK = r_A \,.\, r_S \,.G$ as it is based on random numbers. Also, the adversary will face with the ECCDHP. Therefore, the proposed scheme satisfies the feature of perfect forward secrecy.

5. PERFORMANCE ANALYSIS OF THE PROPOSED SCHEME

In order to evaluate the security and performance of the proposed scheme, comparison of the computation cost of the proposed scheme with related schemes is performed. Table 3 gives review of the performance by computing the time consumed by various operations in each phase. Here TS denote the symmetric key encryption, T_H denote the hash operation, T_E denotes the modulus exponentiation operation, T_{EM} denotes the elliptic curve multiplication, T_A denotes the elliptic curve addition and subtraction, T_X denotes the XOR operation, T_P denotes bilinear pairing operation and TC denotes concatenation operation.

It is analyzed that Islam-Biswas's scheme [2] and C. T. Li's scheme [9] make use of bilinear pairings. It has been found that the cost of the bilinear parings is almost 20 times more than the scalar multiplication over elliptic curve group i.e. $TP \gg TEM$. Also Song [14] uses exponential operation and the time required to implement an exponential operation is approximately 8 times more than one elliptic point multiplication i.e. $T_E \gg T_{EM}$. In the proposed scheme, no bilinear pairing or exponential operation is used. Furthermore [2, 9, 12] makes use of elliptic curve addition/multiplication, which is quite slower than XOR operation thus increasing their final computation cost. The proposed scheme on the other hand makes use of fast XOR operation instead of elliptic curve addition/multiplication which decreases its overall computation cost and therefore improves its efficiency to a great extent. By the above security, functionality and performance comparisons, it is concluded that the proposed scheme requires lower computational cost and provides more security attributes than any of the existing schemes. The proposed scheme thus provides efficiency and security from any type of system intrusion.

6. CONCLUSION

This paper proposes a robust smart card based authentication and key agreement scheme for big data environment. Provision of user anonymity is one of the most desired attribute of big data systems to ensure the privacy of users. The proposed scheme improves all the weaknesses of related schemes and efficiently ensures security of the system. Even if the password of the user and master key of the server are compromised, the proposed scheme still manages to provide a secure session communication between the client and remote server because the session key is generated using random numbers. Furthermore, the proposed scheme requires lower computation cost and provides more security and functionality features than all the existing schemes. In future, the proposed scheme will be implemented in AVISPA and tested against other potential attacks.

7. REFERENCES

[1] L. Lamport, Password authentication with insecure communication, Commun. ACM 24 (11) 770–772 (1981).
[2] Islam, S.H., Biswas, G.P., Design of improved password authentication and update scheme based on elliptic curve

cryptography, Mathematical and Computer Modelling 57 (11–12) 2703–2717 (2013).
[3] Lin, C.L., Hwang T., A password authentication scheme with secure password updating, Computer and Security 22 (1) 68-72 (2003).
[4] D. He, Comments on a password authentication and update scheme based on elliptic curve cryptography, Cryptology EPrint Archive Report 2011/411 (2011).
[5] Wang, R.C., Juang, W.S., Lei, C.L., Robust authentication and key agreement scheme preserving the privacy of secret key, Computer Communications 34 (3) 274–280 (2011).
[6] X.M. Wang, W.F. Zhang, J.S. Zhang, M.K. Khan., Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards, Computer Standards and Interfaces 29 (5) 507–512 (2007).
[7] Debiao He, Shuhua Wu, Jianhua Chen, Note on 'Design of improved password authentication and update scheme based on elliptic curve cryptography', Mathematical and Computer Modelling 55 (3-4) 1661-1664 (2012).
[8] D. Wang, C. G. Ma, L. Shi, and Y. H. Wang, On the security of an improved password authentication scheme based on ECC, in Information Computing and Applications, vol. 7473 of Lecture Notes in Computer Science 181–188 (2012).
[9] C.T. Li, A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card, IET Information Security 7 (1) 3-10 (2012).
[10] Lili Wang, Analysis and Enhancement of a Password Authentication and Update Scheme Based on Elliptic Curve Cryptography, Journal of Applied Mathematics, Volume 2014, Article ID 247836, 11 pages (2014)
[11] P. Qiao, H. Tu, A security enhanced password authentication and update scheme based on elliptic curve cryptography, International Journal of Electronic Security and Digital Forensics 6 (2) 130-139 (2014).
[12] S. Ramesh, Dr.V.Murali Bhaskaran, An Improved Remote User Authentication Scheme with Elliptic Curve Cryptography and Smart Card without using Bilinear Pairings, International Journal of Engineering and Technology (IJET) 5 (6) (2013).
[13] Toan-Thinh Truong, Tran, M.-T, Anh-Duc Duong, Improvement of more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on ECC, 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA) 698-703 (2012).
[14] R. Song, Advanced smart card based password authentication protocol, Computer Standards & Interfaces, Elsevier 32 (4) 321-325 (2010).
[15] Kalra S, Sood SK. Secure authentication scheme for IoT and cloud servers. Pervasive and Mobile Computing. 2015 Dec 31;24:210-23.
[16] Sharma G, Kalra S. Identity based secure authentication scheme based on quantum key distribution for cloud computing. Peer-to-Peer Networking and Applications 1-15 (2016).
[17] Sharma G, Kalra S. A Novel Scheme for Data Security in Cloud Computing using Quantum Cryptography. InProceedings of the International Conference on Advances in Information Communication Technology & Computing 2016 Aug 12 (p. 37). ACM(2016).

Table 3. Comparison of Proposed Protocol with Related Work

Security Characteristics	**Song [14] (2010)**	**Islam and Biswas [2] (2011)**	**C. T. Li [9] (2012)**	**Ramesh and Bhaskaran [12] (2014)**	**Proposed Scheme**
Password guessing attack	No	Yes	Yes	Yes	No
Stolen verifier attack	No	Yes	Yes	Yes	No
Insider attack	Yes	Yes	Yes	Yes	No
Impersonation attack	Yes	Yes	No	Yes	No
Server spoofing attack	Yes	Yes	Yes	Yes	No
Many logged-in users attack	No	No	No	Yes	No
Known session-specific temporary information attack	No	No	No	No	No
Registration Phase	$T_E + 2T_H$	T_{EM}	$2T_{EM}$	$T_{EM} + 3T_H + 4T_X$	$T_{EM} + T_H + 2T_X$
Login & Authentication Phase	$2T_S + 5T_H + T_E$	$2T_S + 4T_H + 6T_{EM} + 2T_P + 2T_A$	$2T_S + 4T_H + 11T_{EM} + 2T_P + 2T_A$	$2T_S + 5T_H + 6T_{EM} + 6T_X + 2T_A$	$2T_S + 4T_H + 7T_{EM} + 2T_X + 4T_C$
Session key generation Phase	$2T_H$	$2T_S + 4T_H + 8T_{EM} + 2T_P + 2T_A$	$2T_S + 4T_H + 13T_{EM} + 2T_P + 2T_A$	$2T_S + 5T_H + 8T_{EM} + 6T_X + 2T_A$	$2T_S + 4T_H + 9T_{EM} + 2T_X + 4T_C$