

Password-Based Authentication System Based on Homomorphic Encryption



Marwan Nayyef, Ali Sagheer
University of Anbar
Iraq
marwanmajeed.n@gmail.com
ali.m.sagheer@gmail.com

ABSTRACT: Many of systems and applications available on the internet require authentication from any person before accessing these systems. therefore, most of the systems based on password for authentication. The biometrics way of authentication came to exist, but it requires hardware and complex mechanisms. Each person has data needs to be fully secured. The password is vulnerable to hacking in the event that the hacker gets the data. This paper presents efficient user Authentication System based on Homomorphic Encryption (ASHE) because of Homomorphic Encryption (HE) performs operations upon encrypted data without decryption, therefore, proposed an algorithm based on HE to encrypt all users' attributes. When the user logins into the system, the login password matches homomorphically with the database. If they are matched, the user is identified as a legitimate user else reject. This achieves better authentication and efficiency and preserves privacy of the user. If the user forgets their password, recovery phase is available. In this phase, the server sends a verification code to the user's email. If it matches, then allows the user to reset the password and also implement Tow Factor Authentication (2FA).

Keywords: Password Based Authentication, Homomorphic Encryption, Authentication System, Elliptic Curve Cryptography

Received: 10 April 2018, Revised 15 May 2018, Accepted 26 May 2018

DOI: 10.6025/jism/2018/8/3/83-93

© 2018 DLINE. All Rights Reserved

1. Introduction

User authentication nowadays is a major problem in an authentication system. And for authentication purpose computer security depends on the password. There are some important characteristics of a password: (1) The Password should be changeable, (2) It should quickly and easily executable, and (3) It should easy to remember.

Authentication is an unavoidable task in security where we use a text password as a security technique, but text passwords are threatened by many attacks [1]. Such as a brute force attack which is a type of password guessing attack and it consists of trying every possible code, combination, or password until you find the correct one. This type of attack may take a long time to

complete. A complex password can make the time for identifying the password by brute force long, the dictionary attack is another type of password guessing attack which uses a dictionary of common words to identify the user's password. Several large-scale password leakages exposed users to an unprecedented risk of disclosure and abuse of their information. This shook the public confidence in the security of the current information infrastructure; the inadequacy of password-based authentication mechanisms is becoming a major concern for the entire information society [2], and online and offline attack, etc. so that many of try for getting information such as username, password, or other user attributes. Also, there is another problem with text-based, password is the difficulty of remembering passwords.

To address the problems with traditional username-password authentication scheme [1], the alternative authentication method such as Authentication Based on Homomorphic Encryption which is proposed to encrypt all user's information to be an available for even service provider, in addition to prevent online and offline attack if the attacker can access the server and get a copy of user's information which was encrypted during the registration phase.

Homomorphic Encryption allows for any person to perform a specific mathematical operation applied on the ciphertext to getting results to be the same results if the same operation has been applied to the original text, this method can provide the confidentiality and the privacy of user's information which are considered sensitive information and no one must access this data, and also exploit this mechanism which is provided from HE for authentication.

Paper structure

2. Related Works

In 2014, Rupali Zamare1 et al, proposed a secure system based on Password-based Authentication to authenticate between the client and server, the key is distributed using DH and ElGamal algorithm, instead of using a single server to store password, this paper proposed a protocol to use two servers to prevent the possibility of obtaining the password. This protocol is efficient because of requires less execution time due to parallel execution and the security against active and passive attack [3].

In 2015 Nishikant S. Burande et al, suggest new authentication system based on password authentication protocol, where two servers adapted to store the password to avoid the breach that might occur. In this paper, ElGamal algorithm and DH are used. The backup services are provided for the purpose of continuing the service, the client information on the server one is kept as a backup on server two and vice versa, if one of the two servers shut down because of some reason, another server must still provide services to the client. This protocol provides a safety against active and passive attack as well [4].

In 2016 K.Suveetha1 et al, proposed banking application for data security, The bank contains a large number of customer information that is confidential and must no one can access that information, so it should preserve the confidentiality of the data, In this paper paillier HE is achieved to apply operation on encrypted banking information because HE allows performing a calculation on ciphertext without using a secret key. In this scheme, security and confidentiality of data are performed [5].

In 2016, Jong-Hyuk Im et al, the use of biometric validation is considered one of the safest methods in the authentication process. In this paper, a palm print authentication scheme was proposed to operate on the Android system, so the biometric data is stored on the remote server in an encrypted form and the matching of the user input to the registered biometric data is computed in an encrypted domain based on pailliar homomorphic encryption. This scheme is performed successfully [6].

In 2017 Marta Gomez-Barrero et al, proposed a security model for verification using biometrics, in this paper, a proposed New verification scheme based on HE for template protection using multi-biometric, where Paillier homomorphic encryption scheme used for encryption, processing, and decrypting data. The only ciphertext is handled using HE verification through computation of original data of biometric and the encrypted template. The results obtained show high accuracy rates [7].

3. Homomorphic Encryption

HE allows anyone to process encrypted data without the need for decryption so that the same value can be obtained from evaluating original and encrypted data. HE concept is shown in the following Figure1. When the user needs to add two numbers such as 5 and 10, the result is 15, the two numbers are encrypted through multiplied with 5, then the sum of the encrypted number is 75 as a result that is stored on the cloud server, the user download data from cloud and recovered the original text [8].

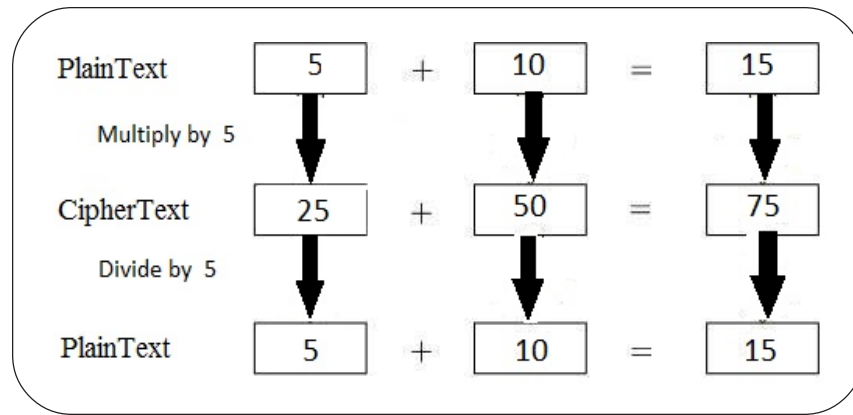


Figure 1. Homomorphic Encryption

3.1 Homomorphic Encryption Functions

There are four algorithms or (primitives) of a public key encryption scheme is that *KeyGen*, *Enc*, and *Dec*, and an additional *Eval*.

- **KeyGen Function:** It is an algorithm in a client which gets security parameter (k) to generate each of the secret key (sk) and public key (pk), $(pk, sk) \leftarrow \text{KeyGen}(k)$.
- **Encryption Function:** Is a random algorithm that produces a ciphertext (c) came from using plaintext and sk , $c \leftarrow \text{Enc}(sk, m)$.
- **Evaluation Functions:** The server use function f for evaluating the ciphertext, and it's done by using f and pk , $\text{Eval}(f, pk, c)$, where $c = (c_1, \dots, c_t)$ and t refer to the number of inputs of the circuit [9]. Therefore $\text{Dec}(sk, \text{Eval}(f, pk, c)) = C(m_1, m_2, \dots, m_t)$, Where C is a computation which perform in the client.
- **Decryption Function:** Is a random algorithm that produces a plaintext (m) came from ciphertext and sk $m \leftarrow \text{Dec}(c, sk)$, and after evaluation, we get the original text as follows $\text{Dec}(sk, \text{Eval}(f, pk, c))$ [10].

3.2 Homomorphic Encryption Properties

Suppose that $m_1, m_2 \in M$ and c_1 and $c_2 \in C$ then $c_1 = \text{Enc}(m_1)$ and $c_2 = \text{Enc}(m_2)$, p is the prime number.

• Additive Homomorphic Encryption

$$\text{Enc}(m_1 + m_2) \bmod p = c_1 + c_2 \bmod p.$$

• Multiplicative Homomorphic Encryption

$$\text{Enc}(m_1 * m_2) \bmod p = c_1 * c_2 \bmod p \quad [11].$$

4. Elliptic Curve Cryptosystem (ECC)

Elliptic curve cryptography is an approach of public-key cryptography, which is based on the structure of algebraic and discrete logarithms on an elliptic curve over finite fields. When elliptic curve (EC) is defined, there are two kinds of a finite field is prime field F_p , where p is a large prime number and binary fields F_{2^m} [12]. It is known the key sizes of ECC are smaller, faster encryption, better security and more efficient for the same level of security compared with other systems of public cryptography (such as RSA) [13].

4.1 Elliptic Curve over Prime Field

Definition: An elliptic curve EC over a prime field F_p is shown in the following equation 1:

$$EC: y^2 \bmod p = x^3 + ax + b \bmod p. \quad (1)$$

Where $a, b \in F_p$ and must satisfies the equation that: $4a^3 + 27b^2 \neq 0 \bmod p$, so the group (F_p) of elliptic curve points $EC(F_p)$ are generated when all points of (x, y) satisfy equation (2.1) of elliptic curves with a point ∞ (called the point at infinity) [13, 14].

4.2 Arithmetic of Elliptic Curve

- Adding and doubling point

Point Addition:

Let point $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, where $x_1 \neq x_2$ means that $P_1 \neq P_2$. (P_1, P_2) belong to $EC(F_p)$ defined in Equation (2.1). The summation of $(P_1 + P_2)$ generates another point $P_3 = (x_3, y_3)$ also must be belong to $EC(F_p)$. Add two points on the elliptic curve are depending on some conditions, shown in the following [15]:

If $P_1 \neq P_2$ with $x_1 = x_2$ and $y_1 \neq y_2$ then $P_1 + P_2 = O$

If $P_1 \neq P_2$ with $x_1 \neq x_2$ then

$$P_1 + P_2 = P_3 = (x_3, y_3)$$

$$x_3 = (\lambda^2 - x_1 - x_2) \bmod p$$

$$y_3 = (\lambda(x_1 - x_3) - y_1) \bmod p$$

Where

$$\lambda = \frac{(y_2 - y_1)}{(x_2 - x_1)} \quad (2)$$

Point Doubling

If $P_1 = P_2$ then

$$P_3 = P_1 + P_1 = 2P_1 = P_3 = (x_3, y_3)$$

$$x_3 = (\lambda^2 - 2x_1) \bmod p$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \bmod p$$

Where

$$\lambda = \frac{3(x_1)^2 + a}{2y_1} \quad (3)$$

4.3 Multiplication over Elliptic Curve

It is one of the basic operation in elliptical curves EC , therefore when multiplying a point of an elliptic curve with an integer k , it means that adding a point with itself in the same number of k times, so certainly suggests the idea of doubling the points to compute $P_2 = k.P_1$ where P_1, P_2 are two points on an $EC(F_p)$ [14].

Definition: (Multiplying an integer number with a Point on an EC):

Let $k \in Z$, and P_1 is a point on an EC , then

$$P_2 = kP_1 = P_1 + P_1 + \dots + P_1 \text{ (k times)}. \quad (4)$$

5. The Proposed System

In proposing system, we depend Homomorphic Encryption property which is performed on the encrypted data for authentication any person who wants to access system resources and also we using OTP (One Time Password) along with an eight-digit pin for login purpose. The OTP is sent to users' e-mail or phone. If they are matching. The Server will allow the user to access his account.

The proposed algorithm based on HE used to encrypt data stored in a cloud environment to dealing with ciphertext without

decryption. And it can provide high speed and security because key generation of algorithm derives from *ECC*, and also depend on Elliptic Curve Discrete Logarithm Problem (*ECDLP*).

1) Key Gen:

- Depend Standard Security Parameter of *ECC* where $SP = (a, b, G, p, n)$
- Select random number r
- Compute $k = r * G = (k_1, k_2)$, where G represents a base point of *EC*.
- Secret key $sk = k_1$.

2) Encryption:

- Make m in range $[0 - p]$.
- $c = m * sk \bmod p$.

3) Decryption:

$$m = c * sk^{-1} \bmod p.$$

4) Evaluation HE:

$$c_1 * c_2 \bmod p = Enc(m_1 * m_2) \bmod p$$

Where $c_1 = m_1 * sk$ and $c_2 = m_2 * sk$

$$c_1 * c_2 \bmod p = (m_1 * sk) * (m_2 * sk)$$

$$c_1 * c_2 \bmod p = m_1 * m_2 * sk^2.$$

The proposed system has four modules: (1) Registration, (2) Authentication, (3) OTP generation and (4) Recovery.

5.1 Registration

The encryption is done by submitting user's data and sending them via a secure channel (SSL) to the server. To preserve user data, all of these data must be encrypted, but except the password (pass) which passed through SHA-256 (d) and then encrypted using the specific algorithm. Finally store all user's data at the server stored in an encrypted form, as shown in the Figure 2.

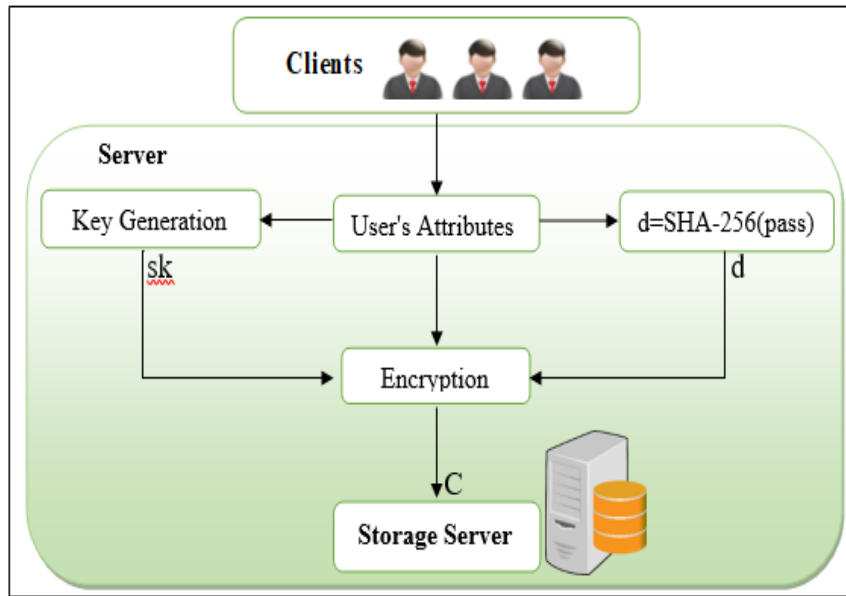


Figure 2. Registration model

An encryption process depends on the secret key (sk), The secret key must be known only by the user. This process prevents anyone even service provider to know the secret key that used by the user for authentication.

Secret Key (sk) is generated by using Password only which passes through SHA-256 to get (d) that is multiplied with a Base Point $G = (x, y)$ to get the secret key point $k = (k_1, k_2)$. In this thesis, k_1 is depends as a secret *key* sk , therefore the password should not be forgotten because if forgotten, the user cannot retrieve his account which is described in figure 3.

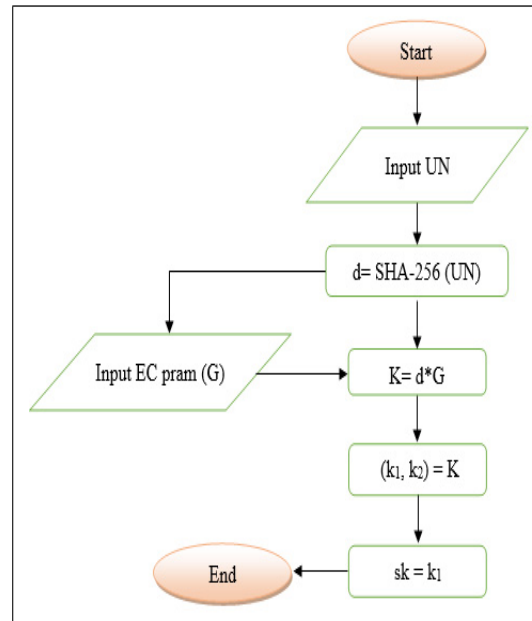


Figure 3. Key Generation Process uses a password

5.2 Login Phase (Authentication)

The process of logging into the system is obtained when the user, who previously registered in the system wants to access system resources by requesting the login page as in Figure 4 and also illustrated in the algorithm1:

Algorithm 1: Authentication process

Goal: Authentication process based on HE

Input: username (UN), password (pass)

Output: true or false

Step 1: The user asks login page and enter **UN** and **pass**

Step 2: **UN** and **pass** are sent via a secure channel (**SSL**) to a **Server**

Step 3: Generate the secret key

Step 4: Computation function of plaintext

4.1 Combine **UN** and **pass**, where $PT = UN \parallel \text{SHA-256}(\text{pass})$

4.2 Compute **Outpt** by performing HE operations on **PT**

Step 5: Computation function of ciphertext

5.1 Retrieve encrypted username and password (**UNct**, **Pct**) from DB then combined, where $C = \text{UNct} \parallel \text{Pct}$

5.2 Compute **Outct** by performing HE operation on **C**

Step 6: Matching results

6.1 If(**Outpt=Outct**)

6.1.1 **Return (true)**

6.2 Else **Return (false)**

5.3 One – Time Password (OTP)

One-time passwords are optimal solutions to provide high security to the system where the use of 64 bits in length, which consider long enough to be secure and short enough to be entered manually by the users. Therefore, the effective role of the dictionary attack and the attack via the communication channel is also a major problem, so the intruder may guess passwords. To solve this problem, Tow Factor Authentication (2FA) based on one-time password was implemented for this purpose. In this way, the attacker cannot access the account even if the password is obtained. In this thesis, adopted the use of the email to receive the verification code sent during the process of login for authentication.

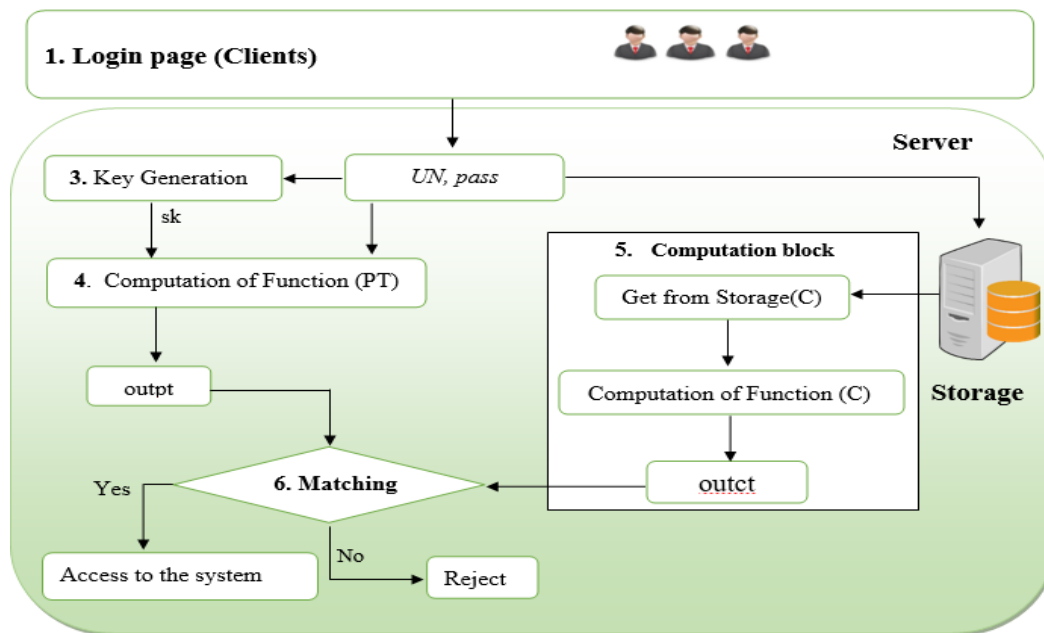


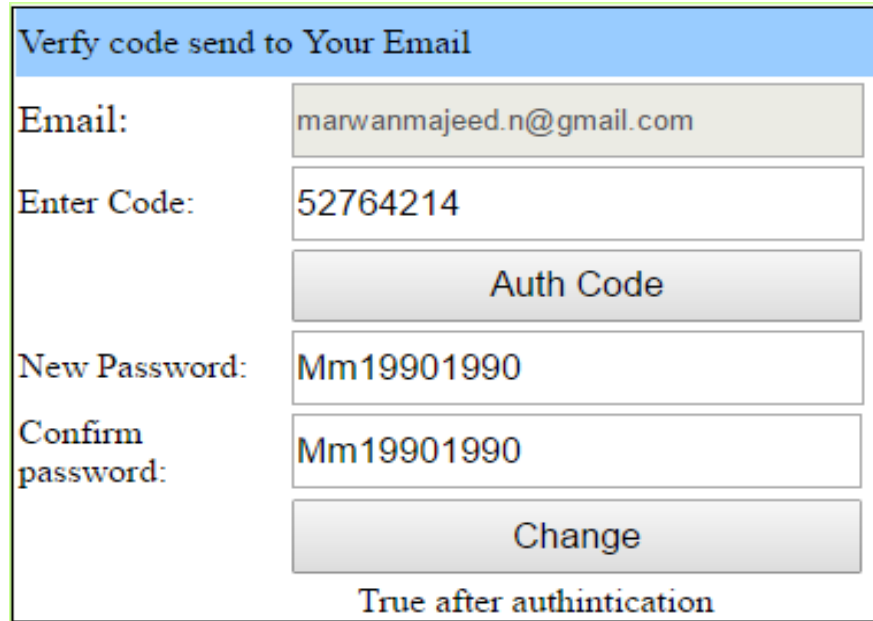
Figure 4. Authentication Model

5.4 Recovery Phase

The recovery phase is utilized when the user forgets his/her password. So the user can retrieve his account if the security key is the username, where it is possible to search for his account homomorphically as in Figure 5 to access the information associated with his/her account. Thus retrieve the encrypted email which was registered in advance, then decrypt the email and displayed to the user. In the same time, sends the verification code to the email until the account ownership is verified. In the end, the user is allowed to reset the password as shown in the following Figure 6.

The screenshot shows a web form titled 'Find Your Account'. It prompts the user to 'Please enter your username to search for your account.' There is a text input field for 'Username:' containing the text 'Marwan90'. Below the input field, it says 'True with time in ms= 42'. At the bottom, there are two buttons: 'Search' and 'Cancel'.

Figure 5. Search for account



Verify code send to Your Email

Email: marwanmajeed.n@gmail.com

Enter Code: 52764214

Auth Code

New Password: Mm19901990

Confirm password: Mm19901990

Change

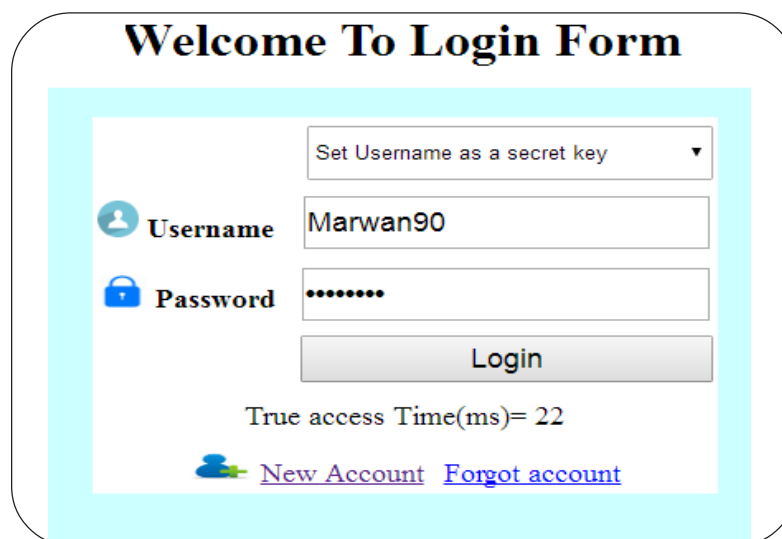
True after authentication

Figure 6. Reset Password

If the user adopted the password as a security key to generate the secret key, recovery of password became impossible, so the user must keep his/her password in a safe manner.

6. Implementation Results

The original text in this system represents the personal attributes that are advanced encrypted during the registration phase. The authentication process takes place during the login phase. At the login phase, the entered username and password transmitted over SSL to the server then evaluated homomorphically to obtain the common value of the original text, then compared with the shared values which is produced by evaluating the encrypted username and password which are stored in the database. In this paper depends 160-bit key length. After matching the two values, the user is allowed to access the system if the two values are identical as in Figure (7 and 8).



Welcome To Login Form

Set Username as a secret key

Username: Marwan90

Password:

Login

True access Time(ms)= 22

[New Account](#) [Forgot account](#)

Figure 7. Valid Login

d=SHA-256 =

k=d.G(gx,gy)=(k1,k2)

k1=

k2=

sk=k1=

Authentication Process based on Homomorphic Encryption (APHE)

Evaluate HE plaintext (outpt) =

Evaluate HE ciphertext (outct) =

if the Two Value **outpt** and **outct** are identical ,the user access to the system else reject

Figure 8. True Authentication based on Homomorphic Encryption

7. Evaluation of Homomorphic Encryption

The use of Homomorphic encryption in the proposed authentication protocol means that it does not require for decryption, whereby it can access the correct information without having to decrypt, and through the algorithm proposed for the system compared with other algorithms such as ElGamal and RSA, note the variation in the execution time in terms of processing time. The following table 1 shows the variation of execution time, Figure 9 plot of the table I. This evaluation of Homomorphic Encryption represents the correct relationship between the original text and its encryption so that we obtain identical results in the case of evaluation of the original text and encryption text at the same time, Therefore, this implementation represents the time required to reach the matching.

No	Byte	Proposed Algorithm	ElGamal	RSA
1.	5	0	0	1
2.	10	0	1	2
3.	30	1	2	3
4.	50	1	2	4
5.	100	2	3	8

Table 1. Evaluation of proposed algorithm, ElGamal and RSA algorithm in ms

If the user enters true username and password, the user can access to the system or application, the authentication phase is happening depends on the Homomorphic Encryption property, many of true authentication of a few users shows in the table 2.

6. Conclusion

Authentication process relies on Homomorphic Encryption which is a proper choice to provide a powerful security for user information. User attributes have been adopted to generate the secret key to encrypt personal attributes and work with them in a way that even the service provider cannot know its contents. Authentication process ensures that the user attributes are processed in encrypted form and do not need to be decrypted during the login phase. The proposed authentication protocol provides confidentiality, privacy and authentication. The encrypted attributes of the user are stored at the remote server and nothing stored on the local server. Depends Two Factor Authentication reduces the opportunity of accessing user account.

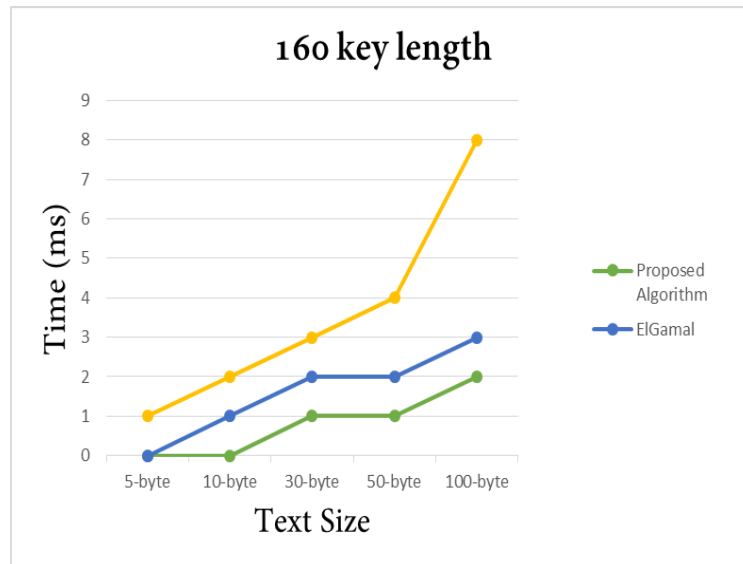


Figure 9. Evaluation Time of Homomorphic Encryption using 160-bit key size

Username (UN)	Password	Execution Time (ms)
Marwan90	19Mmmm90	22
Mohammedsalem70	Msalem123123	30
OmerFalah55	Mfifmf123456	26
AhmedObiadNazzal88	Ahmed19881988	34

Table 2. Execution Time for True Authentication

References

- [1] Shraddha, S., Banne., Shedge., Kishor, N. (2016). CARP: CAPTCHA as A Graphical Password Based Authentication Scheme, *International Journal of Advanced Research in Computer and Communication Engineering*, 5 (1), January 2016.
- [2] Newlin Rajkumar, M., Dhurka, V. (2015). A SECURED PRIVACY AUTHENTICATION WITH RECOVERY, *International Research Journal of Engineering and Technology (IRJET)*, 02 (08), November.
- [3] Rupali Zamare., Prof Rajesh Phursule. (2014). Password Authentication Key Exchange by Two Server Password Only in Web Applications, *International Journal of Recent Development in Engineering and Technology(IJRDET)*, 2 (6), June.
- [4] Nishikant, S., Burande, Prof. Kahate, S.A. (2015). Design Model for Two Server Password Authentication Protocol, *IJCSET*, 11, 5, November.
- [5] Suveetha, K., Manju, T. (2016). Ensuring Confidentiality of Cloud Data using Homomorphic Encryption, *Indian Journal of Science and Technology*, 9 (8), February.
- [6] Im, J. H., Choi, J., Nyang, D., Lee, M. K. (2016). Privacy-Preserving Palm Print Authentication Using Homomorphic Encryption, *In: 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing*, 878-881.
- [7] Marta Gomez-Barrero., Emanuele Maiorana., Javier Galbally., Patrizio Campisi., Julian Fierrez., *Multi-Biometric Template Protection Basedon Homomorphic Encryption, Pattern Recognition*, 67, July, 149-163.
- [8] Kamal Kumar Chauhan., Amit K.S. Sanger., Aja Verma. (2015). Homomorphic Encryption for Data Security in Cloud Computing, *International Conference on Information Technology (ICIT)*, 21-23, December.

- [9] Yatao Yang., Shuang Zhang. (2014). Targeted Fully Homomorphic Encryption Based on a Double Decryption Algorithm for Polynomials, *Tsinghua Science and Technology*, 19 (5), October.
- [10] Lique Chen., Hongmei Ben., Jie Huang. (2014). An Encryption Depth Optimization Scheme for Fully Homomorphic Encryption, International Conference on Identification.
- [11] Jean-Sébastien Coron., David Naccache., Mehdi Tibouchi. (2012). Public Key Compression and Modulus Switching for Fully Homomorphic Encryption over the Integer, EUROCRYPT 2012, LNCS 7237, 446–464.
- [12] Rosy Sunuwar., Suraj Ketan Samal. (2015). *Elgamal Encryption using Elliptic Curve Cryptography*, December 9.
- [13] Sankita J. Patel., Ankit Chouhan., Devesh C. Jinwala, Comparative Evaluation of Elliptic Curve Cryptography Based Homomorphic Encryption Schemes for a Novel Secure Multiparty Computation, *Journal of Information Security*, 2014, 5, 12-18.
- [14] Ammar H. Ali, Ali M. Sagheer. (2017). “Design of a Secure Android Chatting Application using End to End Encryption”, *Journal of Software Engineering & Intelligent Systems (JSEIS)*, 2 (1), April.
- [15] Ziad E. Dawahdeh, Shahrul N. Yaakob., Ali Makki Sagheer., Modified ElGamal. (2015). Elliptic Curve Cryptosystem using Hexadecimal Representation, *Indian Journal of Science and Technology*, 8 (15), July .
- [16] Sagheer, Ali Makki. “Elliptic curves cryptographic techniques, *Signal Processing and Communication Systems (ICSPCS)*, 2012 6th International Conference on. *IEEE*.