ISSN 2047-3338

Chaotic Based Key Management and Public-key Cryptosystem

Mazen Tawfik Mohammed[1], Alaa Eldin Rohiem[2], Ali El-moghazy[3] and A. Z. Ghalwash[4]
[1,2,3]Military technical College, Cairo, Egypt
[4]Helwan University, Egypt

***Abstract*– Public key cryptography is an emerging field whose cryptosystem is based on number theory. However, these cryptosystems suffer from problems such as dealing with large numbers and large prime numbers as well. A recent trend for public key cryptosystems is based on chaotic systems. In this paper a new system for public key cryptosystem and chaotic key management system is introduced. The cryptosystem has been used to provide public-key cryptosystem features such as key-exchange, chaotic key management system and encryption/decryption of intended text. In addition, the proposed cryptosystem protocol solves the man in the middle attack problem since it is based on chaotic management systems. The provided chaotic key management system is based on beta-transformation mapping. The new chaotic key exchange protocol is evaluated against the Diffie-Hellman elliptic curve cryptosystem (DHECC). The proposed system is developed using open source bccrypto-net-1.7 project and C#.net programming language. The test results show that the proposed cryptosystem fills the lack of security gap found in the traditional public-key cryptosystems related to proper key size generation.**

***Index Terms*– Public-Key, Diffie-Hellman Elliptic Curve Cryptosystem (DHECC), Chaotic System and Key-Exchange**

I. INTRODUCTION

PUBLIC key cryptography is a method to exchange secure information such as secret keys via open network. Diffie-Hellman developed in 1976 and published in "New Directions in Cryptography. The protocol allows two users to exchange a secret key over an insecure channel. When user A wants to send a secret message to user B The steps are describe as follow:

1- User A and User B agree upon and make public two numbers g and p, where p is a prime and g is a primitive root mod p.
2- User A chooses a random number (a) and computes $u= g^a$ (mod p) and sends u to user B.
3- User B chooses a random number (b) and computes $v= g^b$ (mod p) and sends u to user A.
4- User A computes $k=v^a= (g^b)^a$ (mod p).
5- User B computes $k=u^b= (g^a)^b$ (mod p).

If user C wants to compute k, then he would need a or b, or user C would solve Discrete Logarithm Problem. Solve for (a) if $u= g^a$ (mod p), given you know u, g and p. with brute force attack and if the prime p had 300 digits and a, b had more than 100 digits; it would take many years to be cracked according to the computational powers of your computer.

Several techniques for public key cryptosystem based chaotic system have been proposed. The advantages and disadvantages of using chaotic systems as cryptosystems are reported in [1]. In this paper we develop a new technique for public-key cryptosystem based on chaotic system that fills the lack of security gap found in the traditional public-key cryptosystem.

The paper is organized as follows: Section 2 discusses the related work. A literature review on public key cryptosystem based chaotic systems is found in Section 3. The proposed scheme for public-key cryptosystem is introduced in section 4. A comparative study with traditional systems is given in section 5. Section 6 includes the conclusion and future work.

II. RELATED WORK

The traditional public-key cryptography has some disadvantages such as key size limitation and the man-in-the-middle attack [2]. Keys are vulnerable to brute force attacks. Generating longer keys will prevent a brute force attack depending on the computing power available to an attacker.

Asymmetric cryptosystems suffer from the key size limitation for an acceptable systems performance that triggers the importance of attempting to come up with asymmetric cryptosystems that are based on chaotic maps in a trial to enhance both security and performance.

Researchers proposed several designs for symmetric and asymmetric cryptosystems based on chaotic maps [3]-[8]. In 2004, a public-key cryptosystem was revealed to be as secure as RSA [9]. In 2005, a key-exchange protocol was announced to asset two communication parties using chaotic dynamics [10]. Later in this year multiple chaotic systems and a set of linear maps for key exchange were utilized and cryptanalyzed [11], [12], [15]. Later the design that of public-key encryption scheme was enhanced by distributed non linear dynamics [13].

The previous public-key cryptosystem based on chaotic systems perform encryption/decryption process depending on addition/subtraction operations that leads to limitation on the length of the message and the weakness in the encryption.

The proposed system enhances the security where the attacker needs high computing power to know the key. The encryption/decryption process is performed in a better way; where the shared key is used as initial condition for chaotic system to produce a random sequence and then apply XOR function between message and the sequence to produce the cipher text.

III. CHAOTIC CRYPTOGRAPHY OVERVIEW

This section introduces the chaotic systems, the logistic map for private key generation, beta-transform used in key exchange and Lorenz system for encryption/decryption.

- Chaotic Systems

Chaos Systems are nonlinear dynamical systems. Depending on the time range they are described by difference equations (discrete-time systems) or differential equations (continuous-time systems). Henon map [16], logistic map [17] and Couple Chaotic Systems Based Pseudo Random Generator (CCSPRBG) [18] are example of discrete-time systems. Rossler system [19] and the Lorenz system [20] are example of continuous-time systems. Chaotic System is sensitive to initial condition, this means that the different initial condition produces different trajectory, the same conditions can produce the same trajectory.

- The Logistic Map System

The logistic map is described in the following equation:

$$x_{n+1} = r x_n (1 - x_n) \quad (1)$$

Where x_{n+1} is the current state variable, x_n is the previous state variable and r is a constant in the range $2 < r < 4$. A small difference in the value of *r* or x_0 can make a huge difference in the outcome of the system after n iteration. There is no equation can determine x value at a specified iteration n even if the initial conditions are known which means the system is unpredictable.

- Beta Transform Cryptosystem

The beta-transformation map system is used to be public key cryptosystem [3].

Let a number B>0, beta transformation is given by as a function $f_B(X_m)=X_{m+1}=BX_m \pmod 1$

where,

$f_B:(0,1)\rightarrow(0,1)$ and m=0,1,2,……..

x(0) is the initial condition.

The beta-transformation map is described in equation (2) Let a_1, $b_1 > 0$ and $a_1 \neq b_1$. the function $F(x_m)$ is defined as:

$$x_{m+1} = F^K(x_m) = [x_m + a_1 x_{m-1}] \pmod 1 \ (k_i = 0)$$
$$= [b_1 x_m + x_{m-1}] \pmod 1 \ (k_i = 1) \quad (2)$$

Where m=0, 1, ….., $X_{-1}=0$, a_1X_{m-1}, b_1X_m are evaluated via the beta-transformation, k is binary string and k_i is a binary value 0 or 1 of position of (i). This system is sensitive to a change in a_1, b_1 parameters. Three kinds of keys are used in this cryptosystem which are a public key, a private key and a common private key. A common private key means shared secret key between two sides.

- Lorenz Chaotic System

The Lorenz chaotic system is a set of nonlinear partial differential equations. It is described in the following equation (3):

$$x_{j+1} = \sigma (y_j - x_j)$$
$$y_{j+1} = \gamma * x_j - y_j - x_j * z_j \quad (3)$$
$$z_{j+1} = x_j * y_j - b * z_j$$

IV. PROPOSED CHAOTIC PUBLIC KEY CRYPTOSYSTEM

The proposed chaotic public-key cryptosystem uses three chaotic maps. The first map is used as generator for private key, the second for shared key and third for encryption. Three logistic maps are used for private key generation and the used chaotic map for the shared key generation is three map of beta-transformation and the used chaotic map for encryption is the Lorenz system.

Private key generation part is described in the following equation (4):

$$\begin{aligned}
x1_{i+1} &= r_1 x1_i (1 - x1_i) && \text{If } x1 > x2 \ k_{i+1} = 1 \\
x2_{i+1} &= r_2 x2_i (1 - x2_i) && \text{If } x2 > x1 \ k_{i+1} = 0 \\
y1_{i+1} &= r_3 y1_i (1 - y1_i) && \text{If } y1 > y2 \ k_{i+1} = 1 \\
y2_{i+1} &= r_4 y2_i (1 - y2_i) && \text{If } y2 > y1 \ k_{i+1} = 0 \\
z1_{i+1} &= r_5 z1_i (1 - z1_i) && \text{If } z1 > z2 \ k_{i+1} = 1 \\
z2_{i+1} &= r_6 z2_i (1 - z2_i) && \text{If } z2 > z1 \ k_{i+1} = 0
\end{aligned} \quad (4)$$

Figure 1 shows the process for private key generation using three logistic maps. The output of private key generator is used in beta-transform.

Shared key generation part use the modified beta-transformation map system is be public key cryptosystem. The modified beta-transformation map is described in the following equation (5):

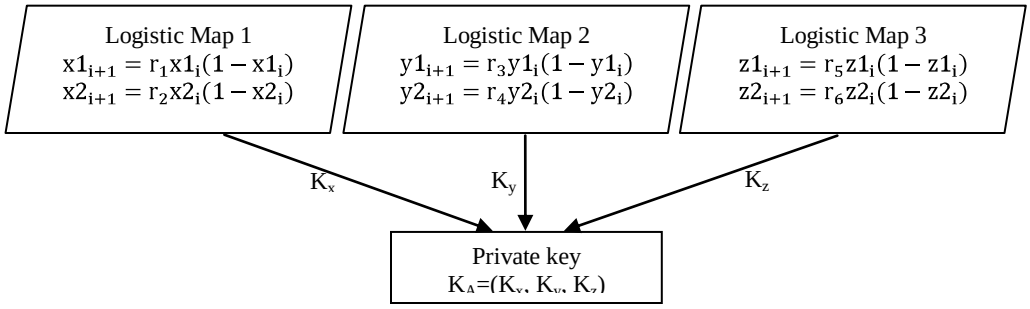
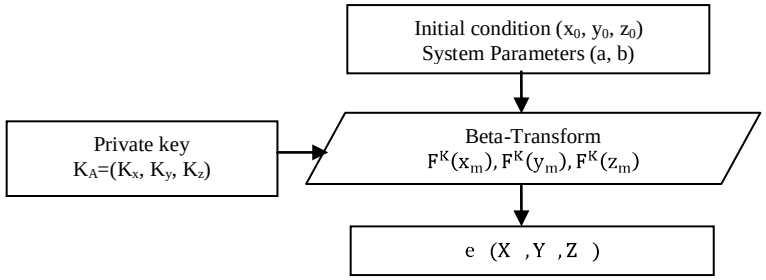


Fig. 1: Private key generator

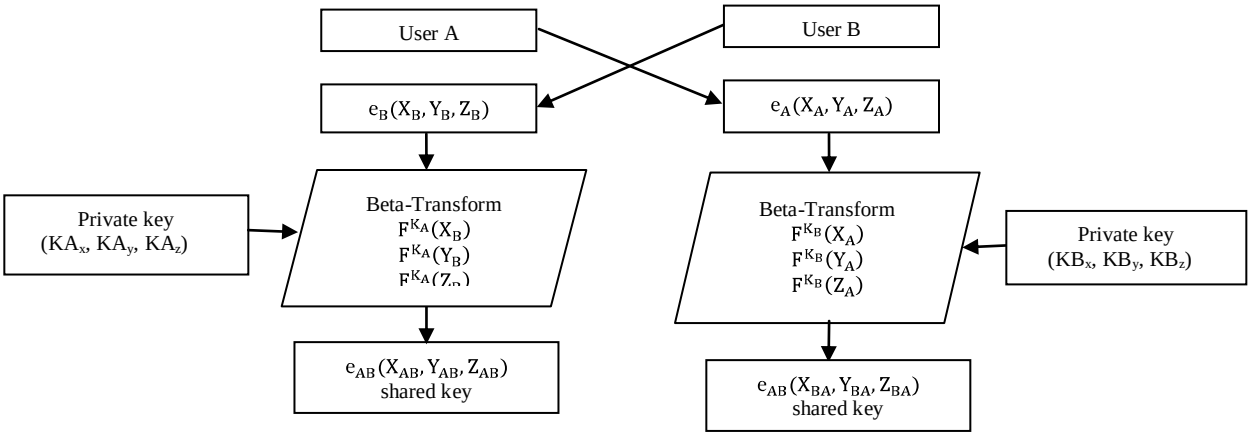
$$
\begin{aligned}
x_{m+1} = F^K(x_m) &= [\, x_m + a_1 x_{m-1}](\mathrm{mod}\ 1) && (k_x(i) = 0)\\
&= [\, b_1 x_m + x_{m-1}](\mathrm{mod}\ 1) && (k_x(i) = 1)\\
y_{m+1} = F^K(y_m) &= [\, y_m + a_2 y_{m-1}](\mathrm{mod}\ 1) && (k_y(i) = 0)\\
&= [\, b_2 y_m + y_{m-1}](\mathrm{mod}\ 1) && (k_y(i) = 1)\\
z_{m+1} = F^K(z_m) &= [\, z_m + a_3 z_{m-1}](\mathrm{mod}\ 1) && (k_z(i) = 0)\\
&= [\, b_3 z_m + z_{m-1}](\mathrm{mod}\ 1) && (k_z(i) = 1)
\end{aligned}
\qquad (5)
$$

The shared key generation steps between two users A and B are:

1- Set initial values x_0, y_0, z_0 for users A and B and set the parameters of three dimension beta-transform (a_1, b_1, a_2, b_2, a_3, b_3) which are common private keys.
2- Generates an N-bit private key1 $K_A=(KA_x, KA_y, KA_z)$ for user A using equation (4).
3- Generates an M-bit private key2 $K_B=(KB_x, KB_y, KB_z)$ for user B using equation (4).
4- Generates a public key $e_A(X_A, Y_A, Z_A) = F^{K_A}(x_0, y_0, z_0)$ using equation (5).
5- Generates a public key $e_B(X_B, Y_B, Z_B) = F^{K_B}(x_0, y_0, z_0)$ using equation (5).
6- Sends e_A to user B.
7- Sends e_B to user A.
8- User A calculate shared key for sender $e_{AB}(X_{AB}, Y_{AB}, Z_{AB}) = F^{K_A}(e_B(X_B, Y_B, Z_B))$
9- User B calculate shared key for sender $e_{BA}(X_{BA}, Y_{BA}, Z_{BA}) = F^{K_B}(e_A(X_A, Y_A, Z_A))$
10-Each side keeps a private key to himself.



(a) Public key generator



(b) Shared key generator

Fig. 2: Public key generator and shared key generator

Figure 2.a shows the process for public key generation and Figure 2.b shows the process for shared key generation.

The Lorenz chaotic system is used for encryption, the cryptosystem is described in the equation (3). Figure 3 shows the encryption part. The output of modified beta-transform is a vector of three values (X_{AB}, Y_{AB}, Z_{AB}), these values are used as initial conditions for Lorenz system (encryption part). Ciphertext is calculated as follow:

Ciphertext=Enc(M, X_{AB}, Y_{AB}, Z_{AB}) =M XOR GenPRN(X_{AB}, Y_{AB}, Z_{AB}).

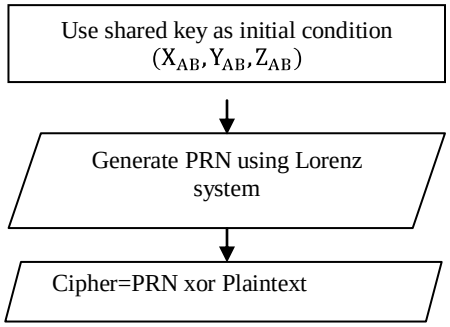


Fig. 3: Encryption Part

The proposed public-key chaotic cryptosystem has the following features:

- The proposed algorithm allows encryption of large length message.
- No padding is required, while most of the other known algorithms always need.
- No relation between key size and plain size
- Fast.
- Unlimited key size

A. Description of Chaotic Key management system

The proposed cryptosystem uses the iteration of modified chaotic map as follow:
At sender side the whole process is executed as follow:

- The user A uses the value K_A as private key.
- The user A calculates the shared key e_{AB} using the private key K_A and public key of user B.
- The encryption functions of user A generate PRN using Lorenz system based on shared key as initial condition then apply XOR function between PRN and plain text.
- The encryption functions are described in equation (6), where C1 are ciphertexts and M is a plaintext.
- The sender sends C1 as a ciphertext to the receiver.

$$C1 = M \text{ XOR } GenPRN(F^{K_A}(e_B(X_B, Y_B, Z_B))) \quad (6)$$
$$= M \text{ XOR } GenPRN(X_{AB}, Y_{AB}, Z_{AB})$$
Where:
$$(X_{AB}, Y_{AB}, Z_{AB}) = e_{AB}(X_{AB}, Y_{AB}, Z_{AB})$$

At receiver side the decryption process is executed as follow:

- The user B uses the value K_B as private key.
- The user B calculates the shared key e_{BA} using the private key K_B and public key of user A.
- The decryption functions of user B generate PRN using Lorenz system based on shared key as initial conditions then apply XOR function between PRN and cipher text.
- The decryption functions are described in equation (7), where C1 are ciphertexts and M is a plaintext.

(7)

$$M = C1 \text{ XOR } GenPRN(F^{K_B}(e_A(X_A, Y_A, Z_A)))$$
$$= M \text{ XOR } GenPRN(F^{K_A}(e_B(X_B, Y_B, Z_B))) \text{ xor } GenPRN(F^{K_B}(e_A(X_A, Y_A, Z_A)))$$
$$= M \text{ XOR } GenPRN(X_{AB}, Y_{AB}, Z_{AB}) \text{ xor } GenPRN(X_{BA}, Y_{BA}, Z_{BA})$$
$$= M$$

Where:

$$(X_{BA}, Y_{BA}, Z_{BA}) = e_{BA}(X_{BA}, Y_{BA}, Z_{BA})$$

Figure 4 shows the whole cryptosystem.

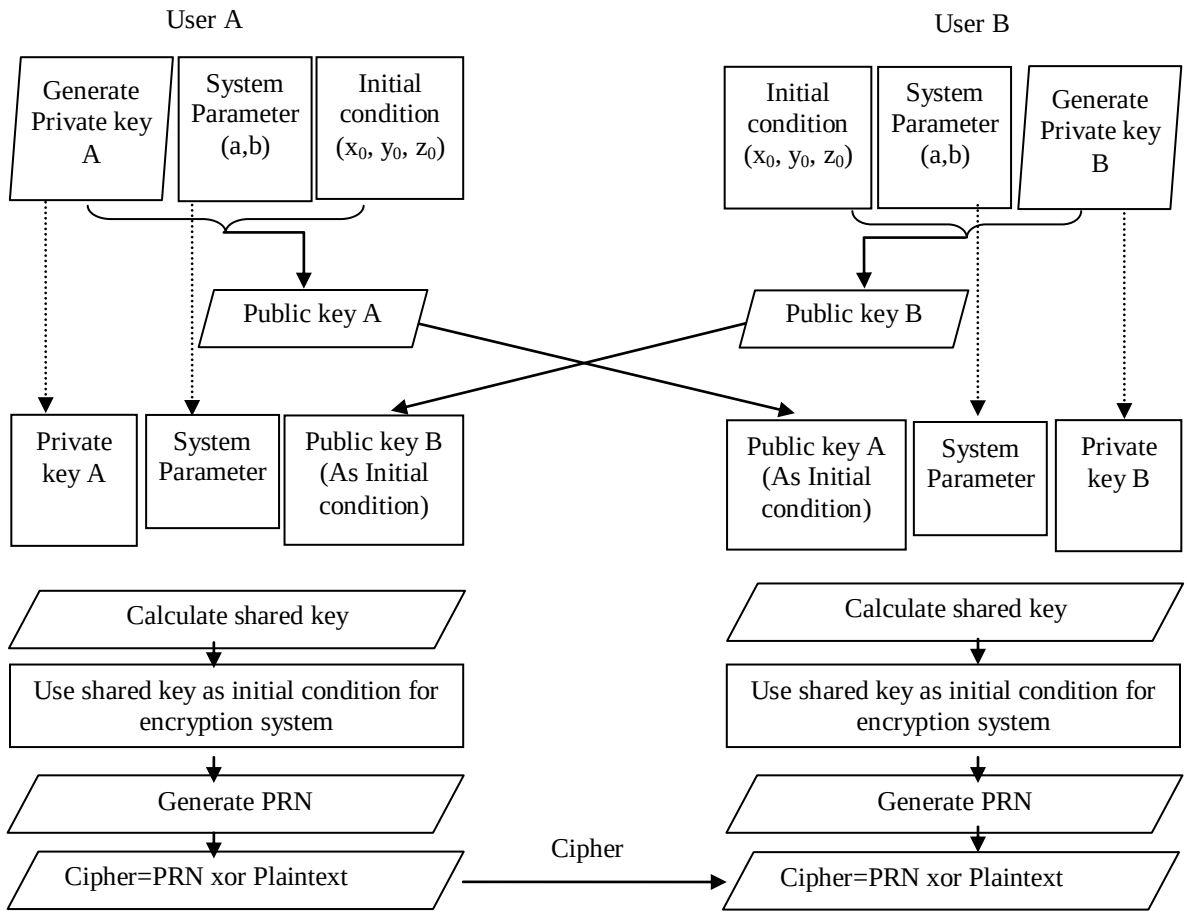


Fig. 4: Block diagram of the proposed Cryptosystem (without solving MITM attack problem)

This cryptosystem suffers from man in the middle attack (MITM). In the next section we propose a solution for this problem.

B. Chaotic Key management system against MITM

This method is used to deal with man in the middle attack [2]. Moreover the encryption complexity is increased with this method due to the use of trusted third party, initial condition and parameters are not known. Before beginning the exchange of keys the clients should be registered at the trusted third party then the trusted third party assign the parameter (initial conditions and system parameters). At sender side the whole process is executed as follow:

- The User A uses the value K_B as a private key.
- The User A calculates the shared key e_{AB} using the private key K_B and public key of User B.
- The encryption functions generate PRN using logistic system based on shared key as initial condition then apply xor function between PRN and plain text.
- The encryption functions are described in equation (8), where C1 are ciphertexts and M is a plaintext.

$$C1 = M \text{ xor } GenPRN(F^{K_B}(e_A(X_A, Y_A, Z_A))) \quad (8)$$
$$= M \text{ xor } GenPRN(X_{AB}, Y_{AB}, Z_{AB})$$
$$\text{Where: } (X_{AB}, Y_{AB}, Z_{AB}) = e_{AB}(X_{AB}, Y_{AB}, Z_{AB})$$

- The sender sends C1 as a ciphertext to the receiver.

At receiver side the decryption process is executed as follow:

- The User B uses the value K_B as a private key.

- The User B calculates the shared key e_{BA} using the private key K_B and public key of User A.
- The decryption functions generate PRN using logistic system based on shared key as initial conditions then apply xor function between PRN and cipher text.
- The decryption functions are described in equation (9), where C1 are ciphertexts and M is a plaintext.

$$
\begin{aligned}
&M = C1 \text{ xor } GenPRN(F^{K_B}(e_A(X_B, Y_B, Z_B))) \\
&= M \text{ xor } GenPRN(F^{K_A}(e_B(X_A, Y_A, Z_A))) \text{ xor } GenPRN(F^{K_B}(e_A(X_B, Y_B, Z_B))) \\
&= M \text{ xor } GenPRN(X_{AB}, Y_{AB}, Z_{AB}) \text{ xor } GenPRN(X_{BA}, Y_{BA}, Z_{BA}) \\
&= M \\
&\text{Where: } (X_{BA}, Y_{BA}, Z_{BA}) = e_{BA}(X_{BA}, Y_{BA}, Z_{BA})
\end{aligned}
\tag{9}
$$

Figure 5 shows the proposed cryptosystem solving the problem of MITM attack.

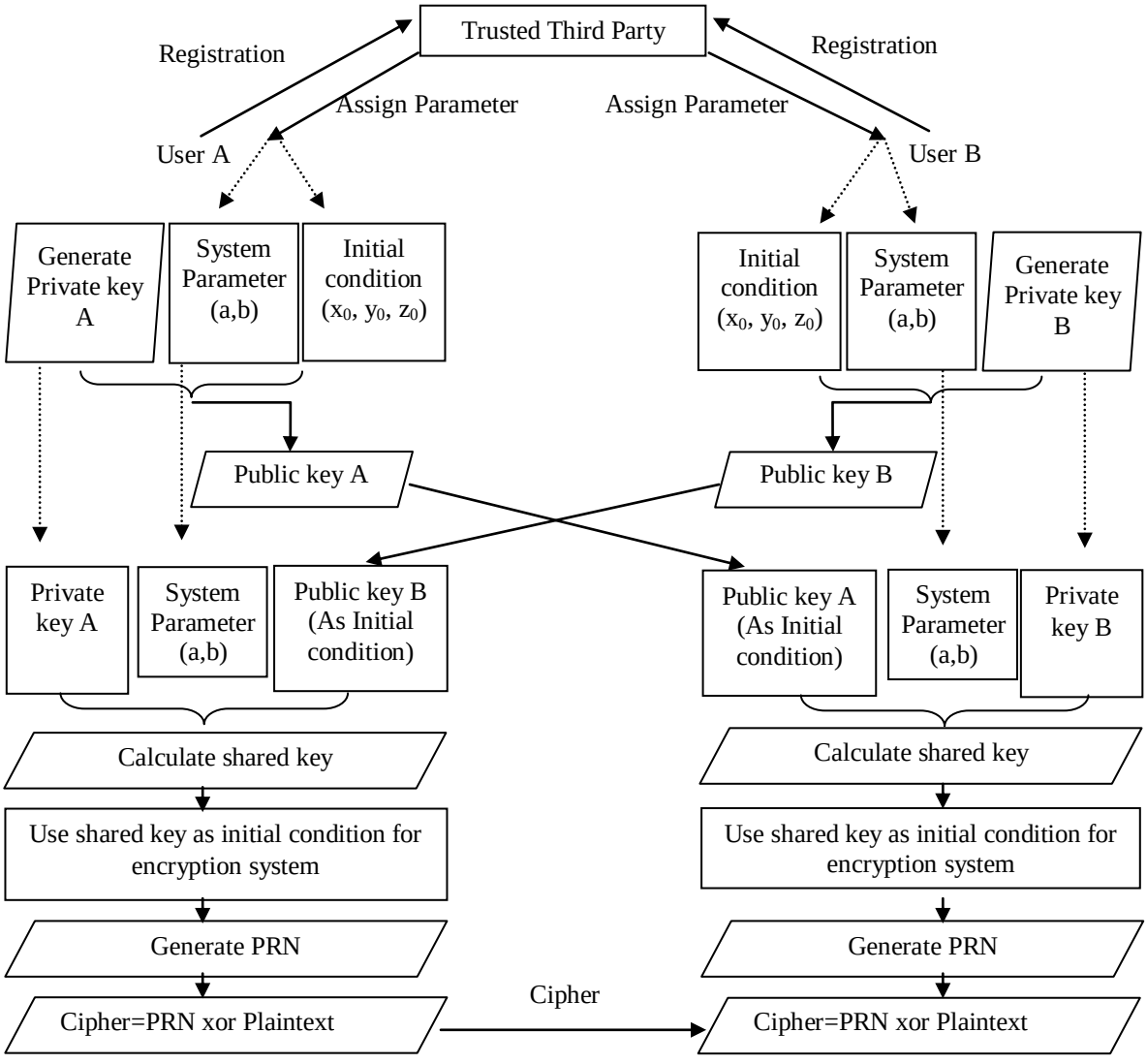


Fig. 5: Block diagram of proposed Cryptosystem (with solving MITM attack problem)

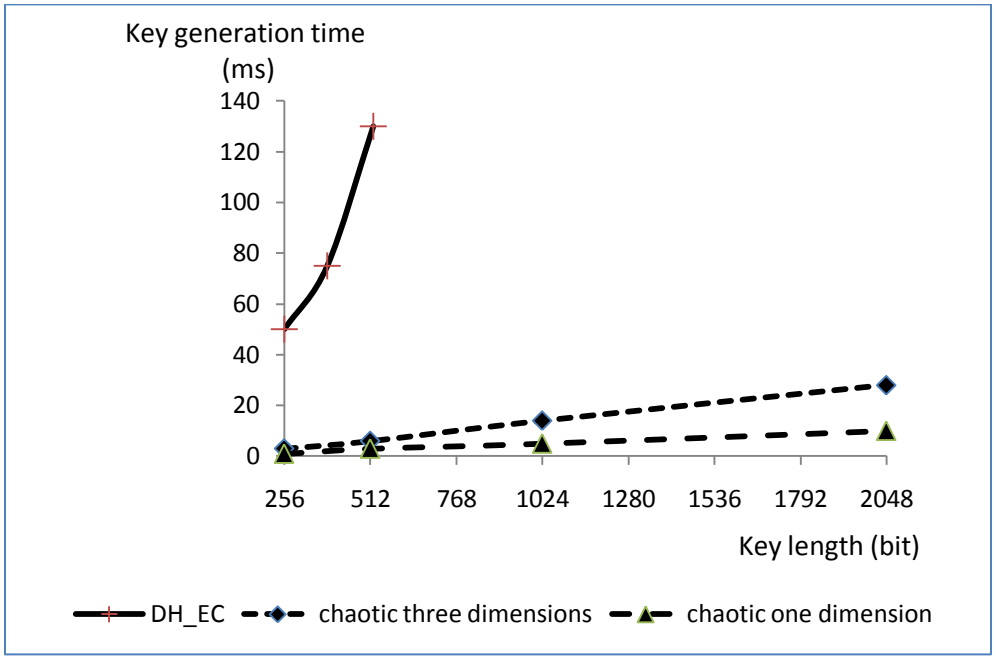


Fig. 6: Generation delay for DH_EC, Chaotic cryptosystem

V. ANALYSIS OF THE PROPOSED CRYPTOSYSTEM

A. Test Bed Configuration

The test bed is composed of two endpoints the first endpoint consists of 2.16 GHz (CPU) with 3 Gbytes RAM and 1 Giga Ethernet for network card, running Windows Vista operating system. The second endpoint consists of 3 GHz (CPU) with 1 Gbytes RAM and 1 Giga Ethernet for network card, running Windows Xp operating system. The proposed scheme is implemented by Visual C#.net.

B. Performance Analysis

The performance is evaluated in terms of generation time for private, public key and shared key considering equivalent key sizes in Diffie-Hellman Elliptic Curve cryptography (DHECC) and chaotic public-key cryptosystem.

1) Key Generation Time

The time of DHEC-Key generation algorithm is measured by using different keys lengths. The generation time of proposed cryptosystem is composed of two parts which are the private key generation time and the public key generation time. The private key generator based on logistic chaotic map, after private key is generated the public key is generated based on modified beta transform system of three dimensions. Figure 6 shows a comparison between DHEC-key generation times, one dimension chaotic-key generation method and three dimensions chaotic-key generation method.

It is clear that the DHEC-key generator is much slower than the key generator of chaotic cryptosystem at the same key size.

C. Security Analysis

The security analysis is performed in terms of key size and brute force attack time.

1) Key and Plaintext Sizes

The proposed cryptosystem has the following parameters:

- The beta-transform system parameters may be public value or common private key.
- Initial conditions (a part of public key).

The key space size consists of private key space. The large key space is good against brute force attack. In the proposed system the private key space is large enough to resist brute force attack. Moreover the tiny change in initial conditions and parameters make the inverse deduction of x(0) from the private key is impossible. The parameters are defined as follows:

- Each parameter in system parameters is 128 bits (or more than 128 bits according to computer capability).
- Each value of initial conditions is 128 bits (or more than 128 bits according to computer capability).
- The length of private key is unlimited length bits.
- The length of message (Plaintext) is unlimited length and the length is variable.
- The length of ciphertext is equal to plaintext length, this means no padding problem.

In the proposed cryptosystem it is very difficult for eavesdroppers to obtain correct private key from the value of initial conditions (X_0,Y_0,Z_0), parameter and public key (X_A,Y_A,Z_A). When the cryptosystem is used with trusted third party then the system parameters and initial conditions are distributed through this party and security level increases.

1) Brute force Attack time

The strength of an asymmetric algorithm such as DHECC and RSA is found in the complexity of computing the inverse of the function used to generate the key. Figure 7 shows the

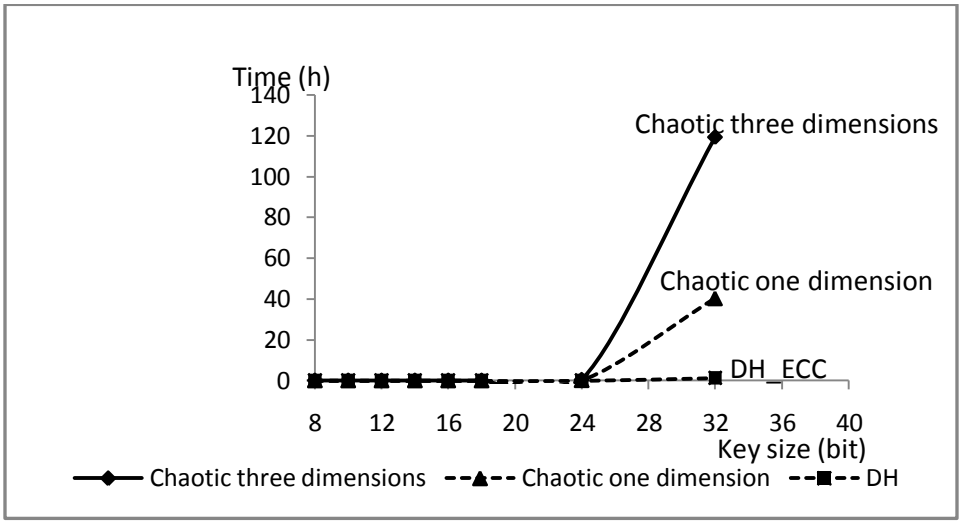


Fig. 7: Brute force attack time for DH_ECC, Chaotic cryptosystems

estimated delay time to perform brute force attack considering DHECC, one dimension chaotic and the proposed system (three dimensions chaotic). The figure shows the superiority of the proposed system.

2) A Chosen Plaintext Attack

A chosen plaintext attack is an attack model for cryptanalysis in which the attacker chooses arbitrary plaintexts to be encrypted and obtains the corresponding ciphertexts. The proposed cryptosystem is effective against chosen plaintext attack. It is difficult for attacker to obtain the correct private key from initial conditions, parameter and X_A, Y_A, Z_A (public key). Moreover the sender and receiver can easily change the private key.

VI. CONCLUSION AND FUTURE WORK

This paper presents a chaotic based key management and public-key cryptosystem to overcome some problems in previous public-key based on chaotic system. The proposed system is based on modified beta-transform map for key exchange and logistic map private key generation and Lorenz for encryption. The proposed system is implemented using C#.net and open source bccrypto-net-1.7 project. Evaluation and comparison with standard mechanism are performed. The results obtained indicate that proposed chaotic public-key cryptosystem enhances performance and security. Future work includes using the proposed public key cryptosystem to enhance Internet security protocols.

REFERENCES

[1] E. Biham, "Cryptanalysis of the chaotic-map cryptosystem," *Proc. Eurocrypt '91*, pp. 532-534, 1991.

[2] Shai Halevi and Hugo Krawczyk, "Public-key cryptography and password protocols," *ACM Transactions on Information and System Security*, pp. 230-268, August 1999.

[3] M. R. K. Ariffin and N. A. Abu, "A Chaos Based Public Key Cryptosystem," *International Journal of Cryptology Research*, pp.149-163, 2009.

[4] L. Kocarev, S.Lian (Eds.), "Chaos-Based Cryptography - Theory, Algorithms and Applications", *Studies in computational Intelligence* vol. 354, Springer, 2011.

[5] L. Kocarev, "Chaos-Based Cryptography: A Brief Overview," *IEEE Circuits and Systems Magazine*, vol. 1, pp. 6-21, 2001.

[6] M. Harada, Y. Nishio and A. Ushida, "A Cryptosystem Using Two Chaotic Maps," *Proceedings of NOLTA'99*, vol. 2, pp. 609-611, 1999.

[7] N. Masuda, K. Aihara, "Cryptosystems with discredited chaotic maps," *IEEE Trans. Circuits and Systems I*, vol. 49, pp. 28-40, 2002.

[8] L. Kocarev, Z. Tasev, "Public-key encryption based on Chebyshev maps," *Proceedings of ISCAS'03*, vol. 3, pp. 28-31, 2003.

[9] L. Kocarev and M. Sterjev, "Public key encryption scheme with chaos," *Chaos*, vol. 14, pp.1078-1081, 2004.

[10] E. Klein, R. Mislovaty, I. Kanter, and W. Kinzel, "Public channel cryptography using chaos synchronization," *Phys. Rev. E.*, vol.72, 2005.

[11] R. Bose, "Novel public key encryption technique based on multiple chaotic systems," *Phys. Rev. Lett.*, vol. 26, 2005.

[12] K. Wang, W. Pei, and L. Zhou, "Security of public key encryption technique based on multiple chaotic systems," *Phys. Lett. A*, vol. 360: pp. 259-262, 2006.

[13] R. Tenny and L. Tsimring, "Additive mixing modulation for public key encryption based on distributed dynamics," *IEEE Trans. Circuits Syst I*, pp. 672-679, 2005.

[14] N. Koblitz, "Elliptic Curve Cryptosystems," *Math. Comp*, vol. 48, pp. 203-209, 1987.

[15] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos, Solitons and Fractals*, vol.37, pp. 669-674, 2008.

[16] J. C. Sprott, "High-Dimensional Dynamics in the Delayed Hénon Map", *Electronic Journal of Theoretical Physics 3*, pp. 19-35, 2006.

[17] J. M. H. Elmirghani, R A. Cryan and S. H. Milner, "Performance of a novel echo cancellation strategy based on chaotic modulated speech," *Proc. SPIE (special issue for chaotic circuits for communication)*, Oct. 1995.

[18] Sh. Li, X. Mou and Y. Cai, "Pseudo-Random Bit Generator Based on Couple Chaotic Systems and its Applications in Stream-Cipher Cryptography", *INDOCRYPT 2001*, LNCS, Springer-Verlag, Berlin, 2001.

[19] O. E. Rossler, "An Equation for Continuous Chaos," *Phys. Lett. A*, vol. 57, no. 5, pp. 397-398, 1976.

[20] C. Sparrow, "The Lorenz Equations in Chaos," *V. Holden. Princeton. University Press, Princeton*, 1986.