Paper: ASAT-15-107-CM

15th International Conference on
***AEROSPACE SCIENCES & AVIATION TECHNOLOGY***,
***ASAT - 15*** – May 28 - 30, 2013, Email: asat@mtc.edu.eg ,
Military Technical College, Kobry Elkobbah, Cairo, Egypt,
Tel: +(202) 24025292 –24036138, Fax: +(202) 22621908

# Chaotic-Based Public Key Cryptosystem for PGP Protocol

M. T. Mohammed[*], A. E. Rohiem[†], A. El-moghazy[†] and A. Z. Ghalwash[§]

**Abstract:** E-mail service is one of the most important internet services. E-mail security protocols are used to provide security services for email systems. It includes confidentiality, authentication, message integrity, non-repudiation. Pretty Good Privacy (PGP) and Secure/Multipurpose Internet Mail Extension (S/MIME) are commonly used email security protocols. The used algorithms in PGP suffer from several problems. The large prime numbers is considered a big problem in public key cryptosystem. The chaotic public key cryptosystem is new trend in cryptography. In this paper a new public key cryptosystem based on chaotic system is introduced. The provided cryptosystem is based on three maps of beta-transformation mapping. The proposed cryptosystem has been used to provide the public key features such as key-exchange, chaotic key management system and encryption/decryption. The new chaotic key exchange protocol is evaluated against the Diffie-Hellman elliptic curve cryptosystem (DHECC). The proposed technique is integrated with the PGP protocol, it is developed using open source bccrypto-net-1.7 project, C#.net programming language and open source for OpenPGP. The test results show that the proposed cryptosystem improved both the security and performance.

**Keywords:** E-mail, PGP, S/MIME, Public-key, DHECC, Chaotic system.

## 1. Introduction

Security is one of the most important issues in mail. An e-mail security is needed to provide confidentiality, data origin authentication, message integrity, non repudiation of origin. The fundamental mechanism for providing security for messages in unsecure network is symmetric and asymmetric algorithms. Symmetric algorithms such as International Data Encryption Algorithm (IDEA) [1], Data Encryption Standard (DES) [2], Triple Data Encryption Standard (TDES) [3], Advanced Encryption Standard (AES) [4]. Asymmetric algorithms such as (RSA) [5] and Diffie-Hellman [6]. The security of a symmetric algorithm is based on the security of the key. The key is used for both encryption and decryption. The key distribution problem is solved by (asymmetric) public key algorithm that uses a pair of keys. A public key is used for encryption and a private is used for decryption. Diffie-Hellman was developed in 1976 and allows two users to exchange a secret key over an insecure channel. With brute force attack and high computational powers of computer the key can be cracked.

In the recent years tremendous interest in the studies of chaos-based cryptography has been observed [8-19, 32]. These studies were greatly encouraged by the increasing number of applications that successfully utilize chaotic systems. The traditional public-key cryptography has some disadvantages such as key size limitation and the man-in-the-middle attack [7].

[*] Syrian Armed Forces, Syria, mazen.mtc@gmail.com .
[†] Egyptian Armed Forces, Cairo, Egypt.
[§] Helwan University, Cairo, Egypt.

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. Researchers proposed several designs for symmetric and asymmetric cryptosystems based on chaotic maps [8-13]. 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 [32]. In 2004, a public-key cryptosystem was revealed to be as secure as RSA [14]. In 2005, a key-exchange protocol was announced to asset two communication parties using chaotic dynamics [15]. Later, multiple chaotic systems and a set of linear maps for key exchange were utilized and cryptanalyzed [16, 17, 18]. The design of public-key encryption scheme was enhanced by distributed non linear dynamics [19]. Other researchers propose enhancements for PGP protocol. In [20] an Enhanced Pretty Good Privacy (EPGP) system with mutual non-repudiation is presented. In this research, a new cryptosystem based on chaotic system for encryption is presented then it is implemented and inserted in PGP protocol.
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 proposed public key cryptosystem is integrated in PGP protocol to enhance the performance and security in PGP protocol. 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. Also in this paper we introduce a novel technique for enhancement security of email protocol based on chaotic systems. The paper is organized as follows: Section 2 discusses email protocols and email security protocol. Chaotic cryptography and a literature review on public key cryptosystem based on chaotic systems are introduced in section 3. The proposed public key cryptosystem for PGP is introduced in section 4. Results and comparison with standard public-key cryptosystem are given in section 5. Section 6 is the conclusion.

## 2. Security of Email

In this section we describe the Email protocols, Email attack and defense, Email security protocol. Typical email architecture contains four elements which are post offices, message transfer agents, gateways and E-mail clients. The outgoing/incoming messages are temporally stored in post offices. Message transfer agents are used for forwarding messages between post offices and to the destination clients. Gateways are used to translate between different e-mail systems, different e-mail addressing schemes and messaging protocols. E-mail clients connect to the post office. IETF publish RFCs about the format of mail messages, email protocols and email security protocols.

### 2.1 Email Protocols

The Simple Mail Transfer Protocol (SMTP) is used to transfer e-mail messages from one host to another, also is used to transfer e-mail messages among separate servers. SMTP is described in RFC 821, it is an Internet standard and uses TCP protocol with port 25. RFC 1869 defines the capability for SMTP service extensions. Extended SMTP (ESMTP) allows new service extensions. SMTP traffic is secured by SSL protocol. The Multipurpose Internet Mail Extensions (MIME) is a standard described in RFC 1521 and RFC 1522. It defines the representation for "complex" message bodies. The "complex" message bodies include messages with embedded graphics or audio clips, messages with file attachments, messages in Japanese or Russian, or signed messages. SMTP cannot be used for languages that are not supported by seven-bit ASCII characters. The binary files, video or audio data cannot be used with SMTP. The Post Office Protocol is used to transfer e-mail messages from a permanent

mailbox to a local computer. The POP client creates a TCP connection to a POP3 server on the mailbox computer. The messages are stored and transferred as text files in RFC 2822 standard format. The computers with a permanent mailbox must run two servers, the first server is a SMTP server that accepts sent mail and adds each incoming message to the user's permanent mailbox, and the second server is a POP3 server allows a user to extract messages from the mailbox and delete them. The Internet Message Access Protocol (IMAP) is a standard protocol for accessing email from your local server. IMAP requires continual access to the server during the time that client is working with its mail. In the (POP3), the mail is saved in mailbox on the server. It is immediately downloaded to your computer and no longer maintained on the server.

### 2.2 Email Common Attacks and Defense

There are different types of attacks on email systems [30-31], the first is spoofing attack which can be used to enable one party to masquerade as another party. The defense for this attack is authentication. Through authentication only trusted users can engage in sessions. The second attack is man in the middle/session hijacking attack, in which an attacker inserts itself between two parties and pretends to be one of the parties, the solution is digital signatures. The third attack is an eavesdropping attack in which an attacker listens to a private communication. The defense for this attack is encryption. The fourth attack is data modification attack in which an attacker changes the data. The defense for this attack is an encrypted message digest. The fifth attack is a dictionary attack in which an attacker uses large set of common used passwords to guess the password. The defense is using strong passwords. The sixth attack is denial of service in which an attacker floods the network or computer with hundred or even million of messages or service request. The defense for this attack is authentication service filtering. Another risk of SMTP is the sending and receiving of viruses and Trojan horses. In this paper we focus on the eavesdropping attack so the vulnerability of secure PGP is related to the used encryption algorithm.

### 2.3 Email Security Protocols

Email security protocol is responsible for protection of email messages and passwords. Email security protocol ensures the privacy of clients. There are some email providers that support a secure mail protocol and others do not support such security protocol. Transport layer Security (TLS) prevents eavesdropping and spoofing between mail servers. It is used to provide endpoint authentication and privacy over the Internet. There are separate protocols which are used (by client) to provide security for email messages, the most common are S/MIME and PGP.

a) Secure/Multipurpose Internet Mail Extension (S/MIME): S/MIME is a security protocol which is designed to provide security for electronic mail. The protocol is an enhancement of the Multipurpose Internet Mail Extension (MIME) protocol. S/MIME is an IETF standard track and defined in several documents such as RFCs (3369, 3370, 3850, 3851, 6211, 5750, 5751, and 4262). S/MIME defines several cryptographic algorithms such as Triple DES (TDES), AES, RC2/40, RSA, Diffe-Hellman, SHA-1, and Digital Signature Standard (DSS). S/MIME version 3 encrypts message content using TDES algorithm in the cipher block chaining mode of operation. It uses the SHA-1 hash algorithm and the Digital Signature Algorithm (DSA) for generating digital signatures. Certificate information is secured with S/MIME to produce a public key cryptography standard (PKCS) object. S/MIME produces the hash value of the message content then it encrypts with the signer's private key.

b) Pretty Good Privacy PGP: Pretty Good Privacy (PGP) is an open source package used for e-mail security. PGP is an IETF standards track and defined in several documents. PGP's advantage is strong encryption worldwide and public source code. While S/MIME is further limited to email only, while the PGP suite includes several other applications.

Table (1) shows the different RFC documents for PGP. Mainly, the PGP protocol is described in RFC 1991 and expanded into OpenPGP in RFC 2440. RFC 2015 describes the different kinds of PGP message which is encapsulated using MIME.

**Table 1. The different RFC documents for PGP**

| RFC | Title |
|---|---|
| 1991 | PGP Message Exchange Formats |
| 2440-4880 | OpenPGP Message Exchange Formats |
| 2015 | MIME Security with Pretty Good Privacy (PGP) |
| 2726 | PGP Authentication for RIPE Database Updates |
| 3156 | MIME Security with OpenPGP |
| 5581 | The Camellia Cipher in OpenPGP |

PGP provides authentication using digital signature, confidentiality using symmetric block encryption, compression using ZIP algorithm and encoding using Rdic-64. Table (2) shows the different algorithms used for security in PGP.

**Table 2. The different algorithms used for security in PGP [21].**

| | | Symmetric algorithm | | Hash Algorithms | |
|---|---|---|---|---|---|
| ID | Description | ID | Description | ID | Description |
| 1 | RSA (encryption/signing) | 0 | No Encryption | 1 | MD5 |
| 2 | RSA (encryption only) | 1 | IDEA | 2 | SHA-1 |
| 3 | RSA (signing only) | 2 | Triple DES | 3 | RIPE-MD/160 |
| 16 | ElGamal (encryption only) | 3 | CAST-128 | 4 | Reserved(double with SHA) |
| 17 | Digital Signature Standard | 4 | Blowfish | 5 | MD2 |
| 18 | Reserved (Elliptic Curve) | 5 | SAFER-SK 128 | 6 | TIGER/192 |
| 19 | Reserved (Elliptic Curve Digital Signature Algorithm) | 6 | Reserved (DES/SK) | 7 | Reserved (HAVAL) |
| 20 | ElGamal (encryption/signing) | 7 | Reserved(AES-128) | 100-110 | Private algorithms |
| 21 | Reserved (Diffie-Hellman) | 8 | Reserved(AES-192) | | |
| 100-110 | Private algorithms | 9 | Reserved(AES-256) | | |
| | | 100-110 | Private algorithms | | |

PGP uses public key algorithms for key exchange. In e-mail security, the encryption/decryption is done using a symmetric-key algorithm, but the secret key to decrypt the message is encrypted with the public key of the receiver and is sent with the message. The packet tag denotes what type of packet the body holds. Table (3) shows some commonly used packet types.

Figure (1) shows encrypted PGP message. Several packets are used to construct encrypted message. The public-key encrypted session key packet is used to encrypt session key using public-key algorithm. The session key is used to encrypt a message.

**Table 3. Commonly used packet types [21]**

| Value | Packet type |
|---|---|
| 1 | Session key packet encrypted using a public key |
| 2 | Signature packet |
| 5 | Private-key packet |
| 6 | Public-key packet |
| 8 | Compressed data packet |
| 9 | Data packet encrypted with a secret key |
| 11 | Literal data packet |
| 13 | User ID packet |

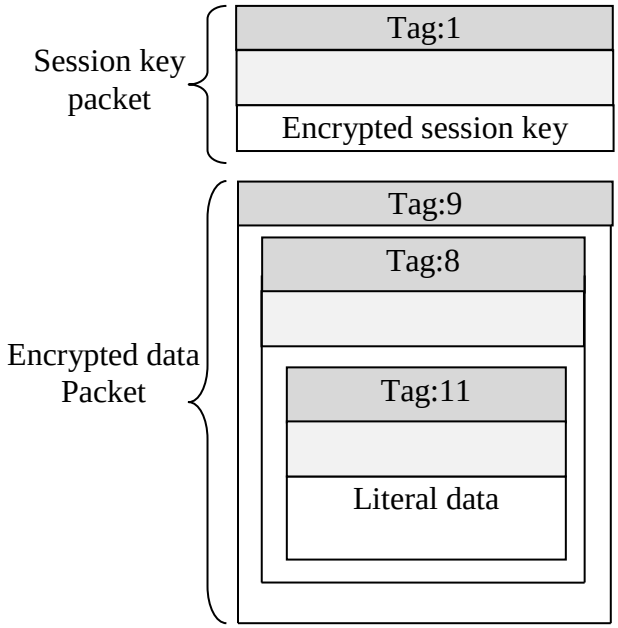



**Fig.1. Encrypted message [21]**

## 2.4 Confidentiality in Standard PGP Algorithm

Symmetric and asymmetric algorithms are used to provide confidentiality for PGP message. Confidentiality is achieved by encrypting the message with a randomly chosen session key and then encrypting the session key with the public key of the recipient. Open PGP message format is described in RFC4880.

a) PGP Message Generation and Reception: Figure (2) shows the diagram for PGP message generation at sender side. PGP uses four types of keys which are one-time session conventional key, public key, private key, passphrase-based conventional key. Private and public keys are stored in two files at each client, these files are called keyrings. Private keys are stored in encrypted form. Decryption key is determined by user-entered passphrase. Random session key is used to encrypt the message. A new session key is required each time a message is encrypted. PGP uses the timing of key strokes and key patterns to generate random numbers. Session key is encrypted using receiver's public key and appended to message. The sending PGP entity performs the following steps:

- Signs the message:
  • PGP gets sender's private key from key ring using its user id as an index.
  • PGP prompts user for passphrase to decrypt private key.
  • PGP constructs the signature component of the message.
- Encrypts the message:
  • PGP generates a session key and encrypts the message.
  • PGP retrieves the receiver public key from the key ring using its user id as an index.
  • PGP constructs session component of message

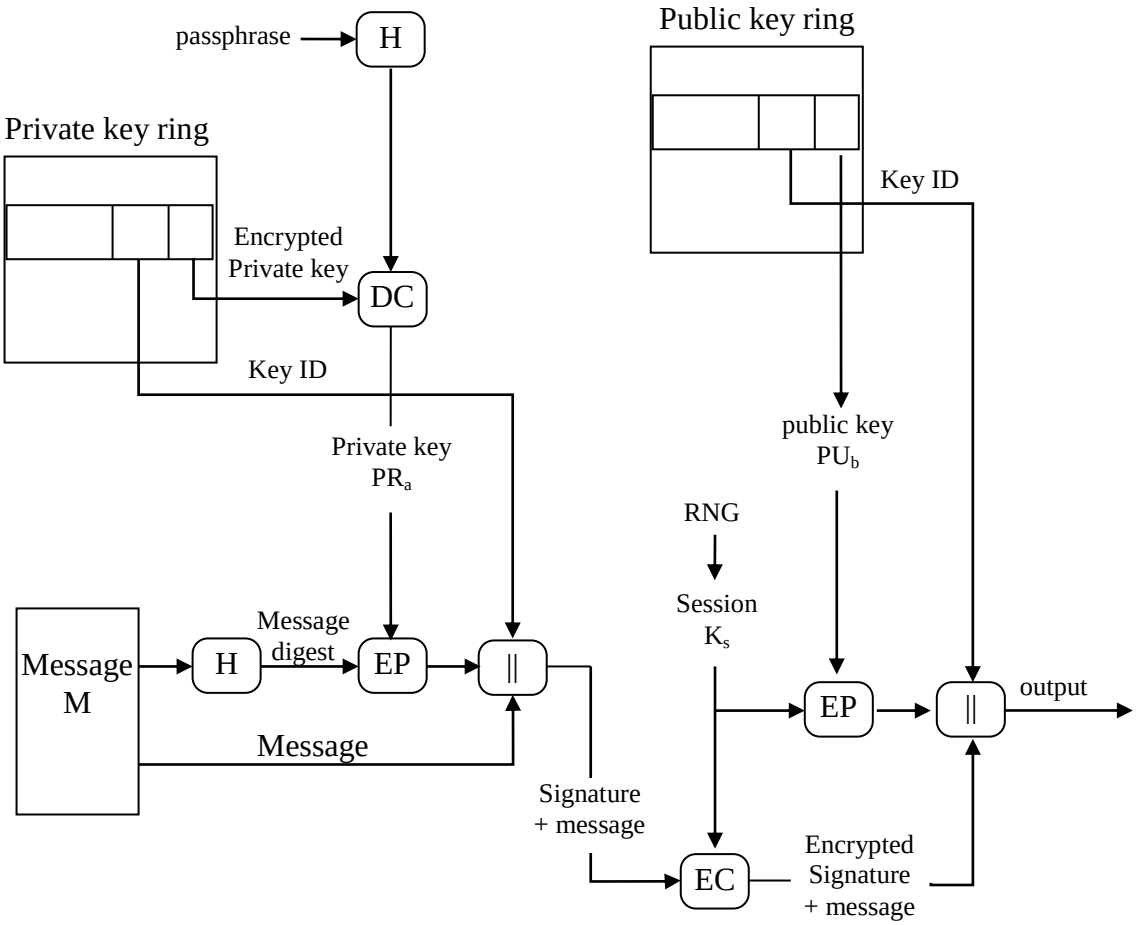



**Fig.2. PGP message generation [22]**
H: hash function; DC: Symmetric decryption process;
EP: Public key Encryption process; EC: Symmetric encryption process;

Figure (3) shows the diagram for PGP message generation at receiver side. Asymmetric and symmetric key cryptosystems are combined in this way to provide security for key exchange and then efficiency for encryption. The session key k is used only to encrypt message m and is not stored for any length of time. The schemes for authentication and confidentiality can be combined so that receiver can sign a confidential message which is encrypted before transmission. The receiving PGP entity performs the following steps:

- Decrypting the message: PGP get private key from private-key ring using Key ID field in session key component of message as an index then PGP prompts user for passphrase to decrypt private key. After that PGP recovers the session key and decrypts the message.

- Authenticating the message: PGP retrieves the sender's public key from the public-key ring using the Key ID field in the signature key component as index. Then PGP recovers the transmitted message digest. After that PGP computes the message for the received message and compares it to the transmitted version for authentication.

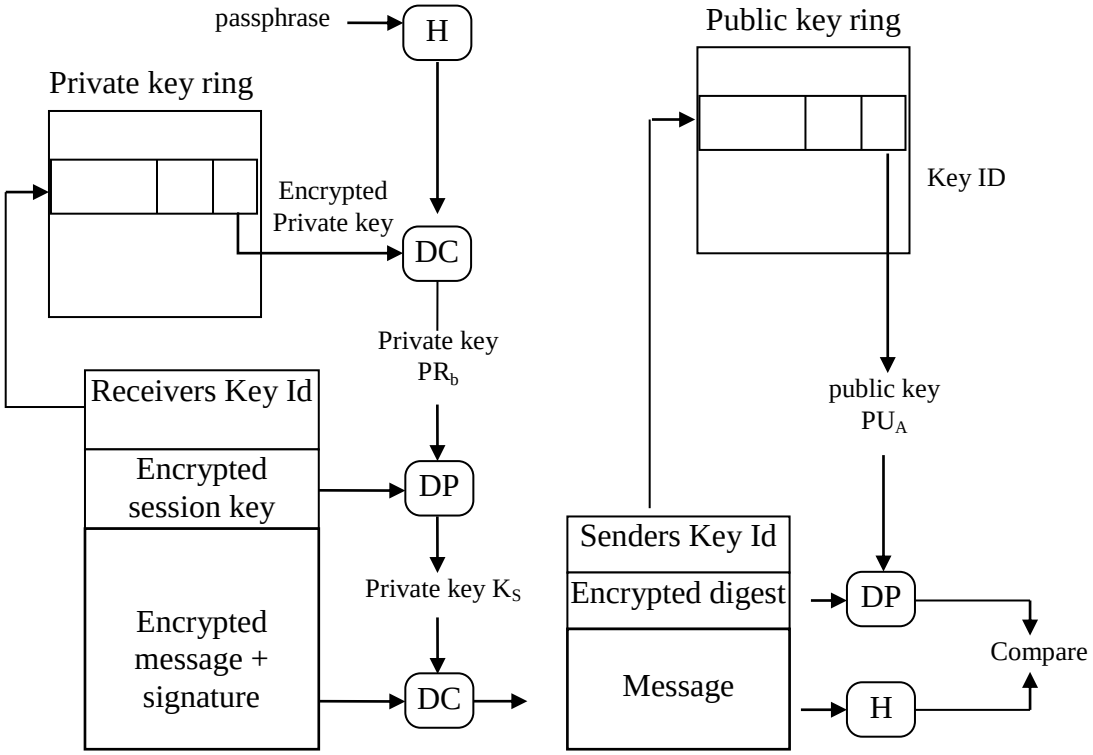



**Fig.3. PGP message reception [22]**
H: hash function; DC: Symmetric decryption process;
DP: Public key decryption process;

## 3. Chaotic Cryptography

In the recent years tremendous interest in the studies of chaos-based cryptography has been observed. These studies were greatly encouraged by the increasing number of applications that successfully utilize chaotic systems. 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.

### 3.1 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 [23], logistic map [24] and Couple Chaotic Systems Based Pseudo Random Generator (CCSPRBG) [25] are example of discrete-time systems. Rossler system [26] and the Lorenz system [27] are example of the 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. Lyapunov Exponents is used to define if the system has chaotic behavior or not, if the system is chaotic the difference between two trajectories with close initial condition will exponentially increase after a very short time. The difference is defined using the equation (1) [28]:

$$d_t=d_0 2^{\lambda t} \quad (1)$$

where:

$d_0$: is initial distance.
$d_t$: is the distance at t time.
$\lambda$: is Lyapunov Exponents.

The value of Lyapunov Exponents ($\lambda$) is solved by averaging the points, using the equation (2) [28]:

$$\lambda = \frac{1}{t_N - t_0} \sum_{k=1}^{N} \log_2 \frac{d(t_k)}{d(t_{k-1})} \quad (2)$$

where:

N: the number of points.
$t_0$: time at initial point.
$t_N$: time at point N.
$d(t_k)$: the distance at point k.
$d(t_{k-1})$: the distance at point k-1.

If $\lambda>0$ the system is considered as chaotic.

a) The Logistic Map System:

The logistic map is described in the following equation:

$$x_{n+1} = r x_n (1 - x_n) \quad (3)$$

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 iterations. 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.

b) Beta transforms Cryptosystem:

The beta-transformation map system is used to be public key cryptosystem [8]. Let a number B>0, beta transformation is given by as a function $fB(Xm)=Xm+1=BXm \pmod 1$ 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 (4) Let $a_1$, $b_1 >0$ and $a_1 \neq b_1$. the function $F(x_m)$ is defined as

$$\begin{aligned} x_{m+1} = F^K(x_m) &= [x_m + a_1 x_{m-1}] \pmod 1 \quad (k_i = 0) \\ &= [b_1 x_m + x_{m-1}] \pmod 1 \quad (k_i = 1) \end{aligned} \quad (4)$$

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.

c) The Lorenz System:

The purpose of Lorenz [1963] was to create and analyze a model for the unpredictable behavior of the weather. By greatly simplifying and truncating a set of nonlinear partial differential equations, he obtained the following system of ordinary differential equations (5) [27]:

$$x' = \sigma (y - x)$$
$$y' = \gamma x - y - xz \quad (5)$$
$$z' = xy - bz$$

A typical Lorenz chaotic attractor can be obtained by setting the parameters σ=10, γ=28, and b=8/3 with initial conditions $(x_0; y_0; z_0)=(1; 1; 1)$. Note that the Lorenz equation has three parameters and two nonlinearities (*xz* and *xy*), each of which is a function of two variables. The theoretical Lyapunov exponent for the Lorenz system is equal to1.5 [28].

## 4. 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 by the following equation (6):

$$\begin{aligned} x1_{i+1} &= r_1 x1_i(1 - x1_i) \\ x2_{i+1} &= r_2 x2_i(1 - x2_i) \end{aligned} \quad \begin{aligned} &\text{If } x1 > x2\ k_{i+1} = 1 \\ &\text{If } x2 > x1\ k_{i+1} = 0 \end{aligned}$$

$$\begin{aligned} y1_{i+1} &= r_3 y1_i(1 - y1_i) \\ y2_{i+1} &= r_4 y2_i(1 - y2_i) \end{aligned} \quad \begin{aligned} &\text{If } y1 > y2\ k_{i+1} = 1 \\ &\text{If } y2 > y1\ k_{i+1} = 0 \end{aligned} \quad (6)$$

$$\begin{aligned} z1_{i+1} &= r_5 z1_i(1 - z1_i) \\ z2_{i+1} &= r_6 z2_i(1 - z2_i) \end{aligned} \quad \begin{aligned} &\text{If } z1 > z2\ k_{i+1} = 1 \\ &\text{If } z2 > z1\ k_{i+1} = 0 \end{aligned}$$

Figure (4) shows the process for private key generation using three logistic maps. The output of private key generator is used in beta-transform.

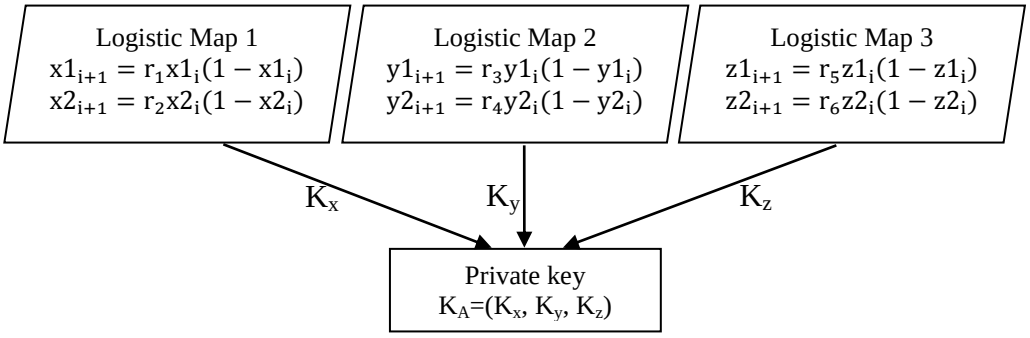



**Fig.4. private key generator**

Shared key generation part use the modified beta-transformation map system to be public key cryptosystem. The modified beta-transformation map is described by the following equation (7):

$$\begin{aligned} x_{m+1} = F^K(x_m) &= [x_m + a_1 x_{m-1}] (\text{mod } 1) \\ &= [b_1 x_m + x_{m-1}] (\text{mod } 1) \end{aligned} \quad \begin{aligned} &(k_x(i) = 0) \\ &(k_x(i) = 1) \end{aligned}$$

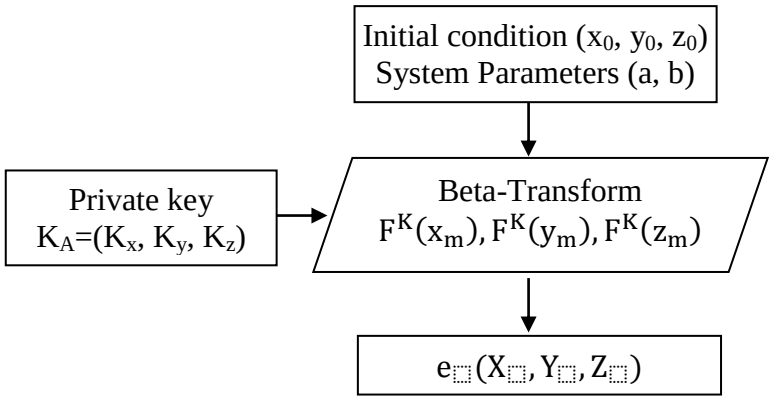
$$\begin{aligned} y_{m+1} = F^K(y_m) &= [y_m + a_2 y_{m-1}] (\text{mod } 1) \\ &= [b_2 y_m + y_{m-1}] (\text{mod } 1) \end{aligned} \quad \begin{aligned} &(k_y(i) = 0) \\ &(k_y(i) = 1) \end{aligned} \quad (7)$$

$$\begin{aligned} z_{m+1} = F^K(z_m) &= [z_m + a_3 z_{m-1}] (\text{mod } 1) \\ &= [b_3 z_m + z_{m-1}] (\text{mod } 1) \end{aligned} \quad \begin{aligned} &(k_z(i) = 0) \\ &(k_z(i) = 1) \end{aligned}$$

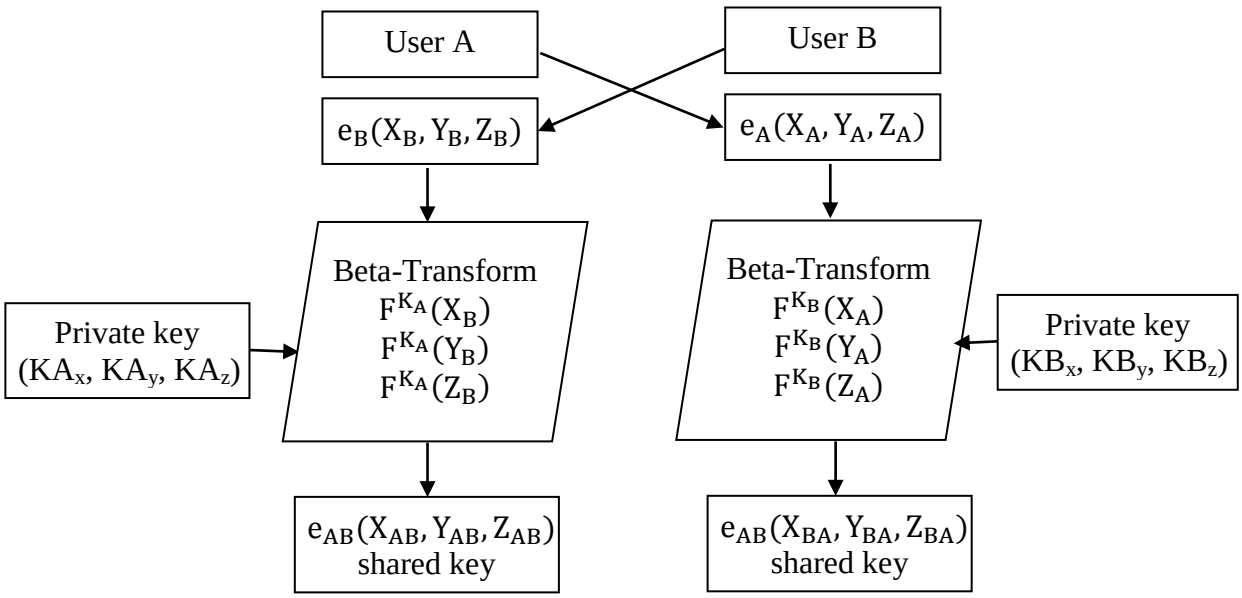
The shared key generation steps between two users A and B are:

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- User A generates a N-bit private key1 KA=(KAx, KAy, KAz) using Eq(6).
3- User B generates a M-bit private key2 KB=(KBx, KBy, KBz) using Eq(6).
4- User A generates a public key $e_A(X_A, Y_A, Z_A) = F^{K_A}(x_0, y_0, z_0)$ using Eq (7).
5- User B generates a public key $e_B(X_B, Y_B, Z_B) = F^{K_B}(x_0, y_0, z_0)$ using Eq (7).
6- User A sends $e_A$ to user B.
7- User B sends $e_B$ to user A.
8- User A calculates shared key $e_{AB}(X_{AB}, Y_{AB}, Z_{AB}) = F^{K_A}(e_B(X_B, Y_B, Z_B))$
9- User B calculates shared key $e_{BA}(X_{BA}, Y_{BA}, Z_{BA}) = F^{K_B}(e_A(X_A, Y_A, Z_A))$
10- Each side keeps a private key to himself.

Figure (5.a) shows the process for public key generation and figure (5.b) shows the process for shared key generation.




(a) Public key generator




(b) Shared key generator

**Fig.5. Public key generator and shared key generator**

The Lorenz chaotic system is used for encryption, the cryptosystem is described in the equation (5). Figure (6) 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($X_{AB}, Y_{AB}, Z_{AB}$).

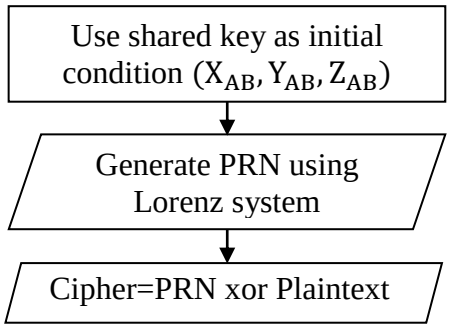



**Fig.6. Encryption Part**

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

### 4.1 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 (8), where C1 are ciphertexts and M is a plaintext.
- The sender sends C1 as a ciphertext to the receiver.

$$C1 = M \text{ XOR } GenPRN(F^{K_A}(e_B(X_B, Y_B, Z_B))) \quad (8)$$
$$= M \text{ XOR } GenPRN(X_{AB}, Y_{AB}, Z_{AB})$$

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 (9), where C1 are ciphertexts and M is a plaintext.

$$M = C1 \text{ XOR } GenPRN(F^{K_B}(e_A(X_A, Y_A, Z_A))) \quad (9)$$
$$= M \text{ XOR } GenPRN(F^{K_A}(e_B(X_B, Y_B, Z_B))) \text{ xor } GenPRN(F^{K_B}(e_A(X_A, Y_A, Z_A)))$$
$$= M \text{ XOR } GenPRN(X_{AB}, Y_{AB}, Z_{AB}) \text{ xor } GenPRN(X_{BA}, Y_{BA}, Z_{BA})$$
$$= M$$
where: $(X_{BA}, Y_{BA}, Z_{BA}) = e_{BA}(X_{BA}, Y_{BA}, Z_{BA})$

Figure (7) shows the whole cryptosystem.

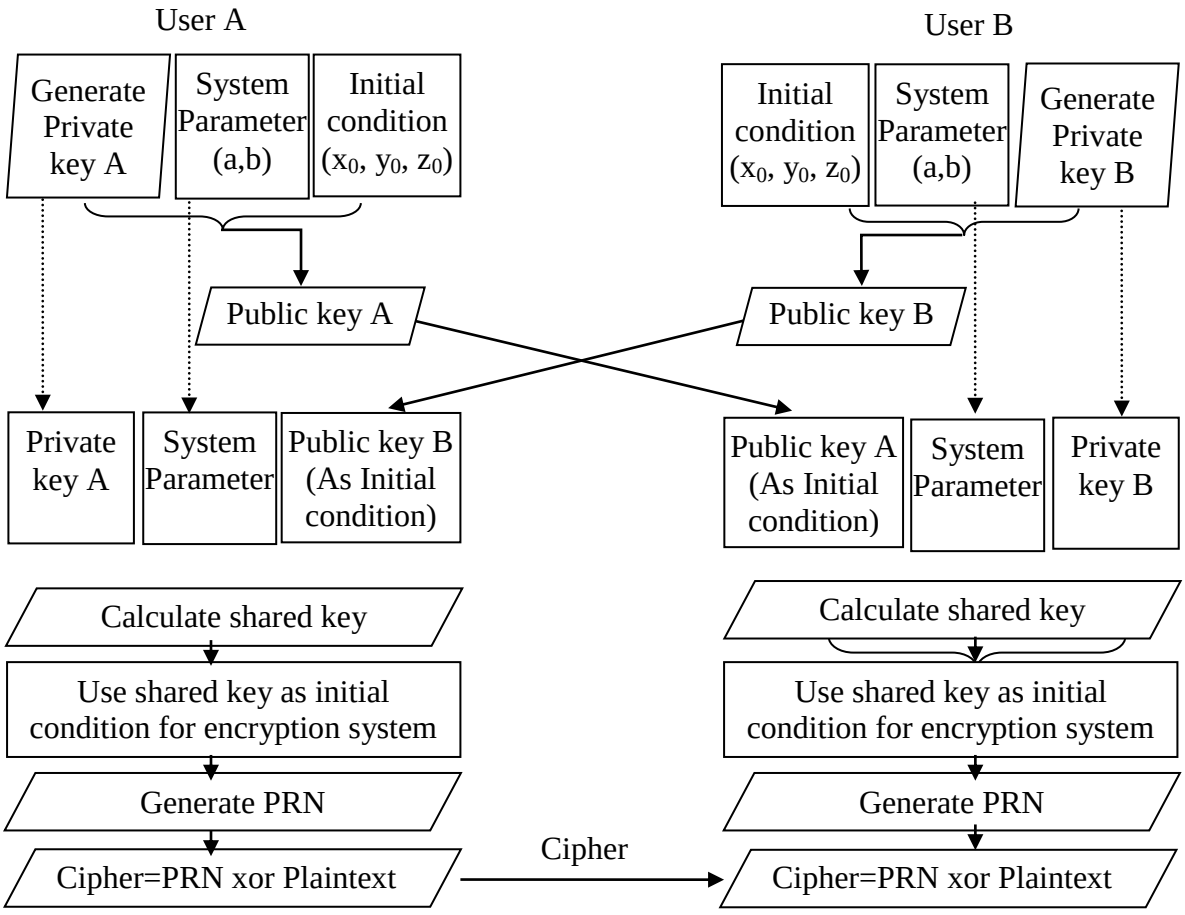



**Fig.7. Block diagram of the proposed Cryptosystem**

### 4.2 Implementing public key Chaotic cryptosystem in PGP

Table (4) contains the commonly used asymmetric key algorithm and the corresponding ID for each algorithm in PGP protocol. The range from 100 to 110 is used from designers for private algorithms. The new identifier for symmetric chaotic encryption is 100.

Public-key packet contains the public information of a key as shown in figure (8). They do not contain information about the owner, so it is typically found combined with ID packets and signatures.

**Table 4. Public key Chaotic cryptosystem in PGP**

| ID | Description |
|---|---|
| 1 | RSA (encryption or signing) |
| 2 | RSA (for encryption only) |
| 3 | RSA (for signing only) |
| 16 | ElGamal (encryption only) |
| 17 | DSS |
| 18 | Reserved for elliptic curve |
| 19 | Reserved for ECDSA |
| 20 | ElGamal (for encryption or signing) |
| 21 | Reserved for Diffie-Hellman |
| 100 | Public key Chaotic |
| 101-110 | Private algorithms |

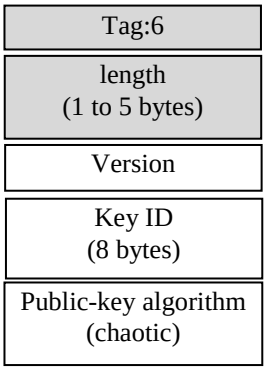



**Fig.8. Public-key packet**

## 5. Analysis of The proposed Cryptosystem:

The proposed Chaotic Cryptosystem is evaluated in terms of performance and security for diffie-hellman algorithms vs. chaotic.

### 5.1 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. The implementation of proposed cryptosystem is tested on test network. The network consists of three computers, two computers for email message transmission and the third for email server. Also the third computer is used for monitoring email message traffic.The test bed is composed of three 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 third endpoint consists of 3 GHz (CPU) with 1 Gbytes RAM and 1 Giga Ethernet for network card, running Windows Xp operating system.

The proposed encryption system is implemented using sharp privacy project for open PGP, open source bccrypto-net-1.7 project and Visual C#.net 2008. The implemented software

includes a set of modules; the modules are either COTS (Commercial Off-The-Shelf) or developed specially for the purpose of the tests.

a) *COTS*: There are three modules which are symmetric module, asymmetric module, PGP module and Email packet module.
   - Symmetric module: includes AES, DES, and TDES.
   - Asymmetric module: includes RSA and Diffie-Hellman.
   - PGP module: sending and receiving PGP packets.
   - Email packet module: forming PGP packets.

b) Developed for the purpose of the tests: Microsoft project doesn't contain the chaotic module. We implement the chaotic module and insert it in the project. Chaotic module contains the following functions:
   - Chaotic public key cryptosystem.

### 5.2 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.

The time of DHEC-Key generation algorithm is measured 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 is 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 (9) shows a comparison between DHEC-key generation times, one dimension chaotic-key generation method and three dimensions chaotic-key generation method.

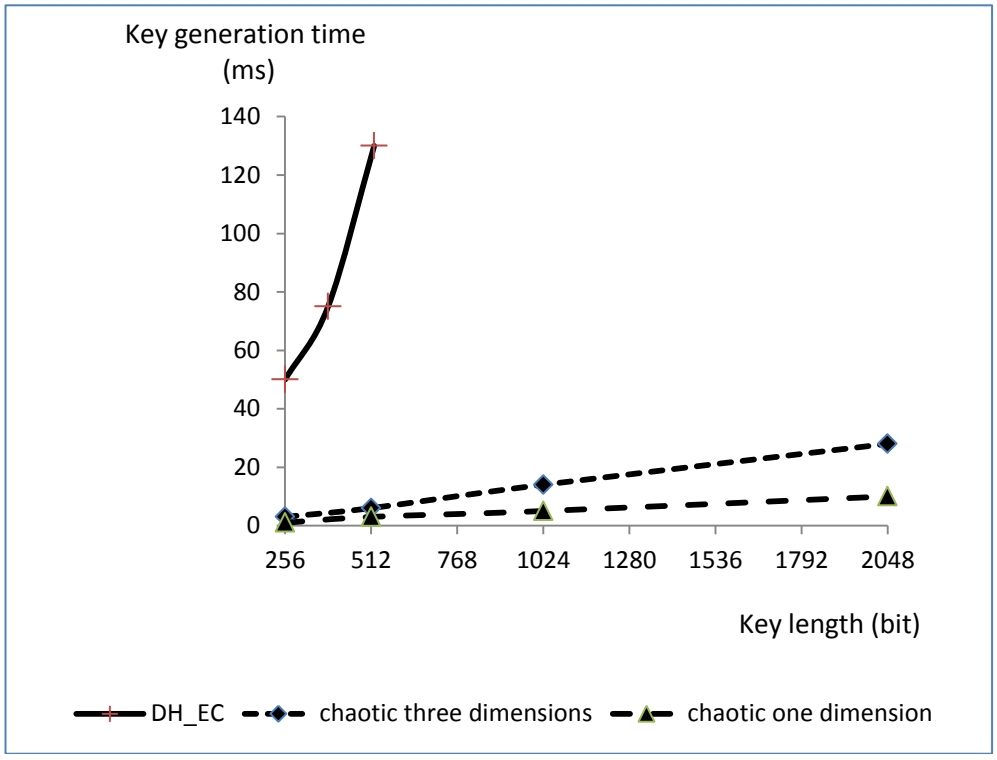



**Fig.9. Generation delay for DH_EC , Chaotic cryptosystem**

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

### 5.3 Security analysis:

The security analysis is performed in terms of key, plaintext sizes and brute force attack time.

a) 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.

b) 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 (10) shows the 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.

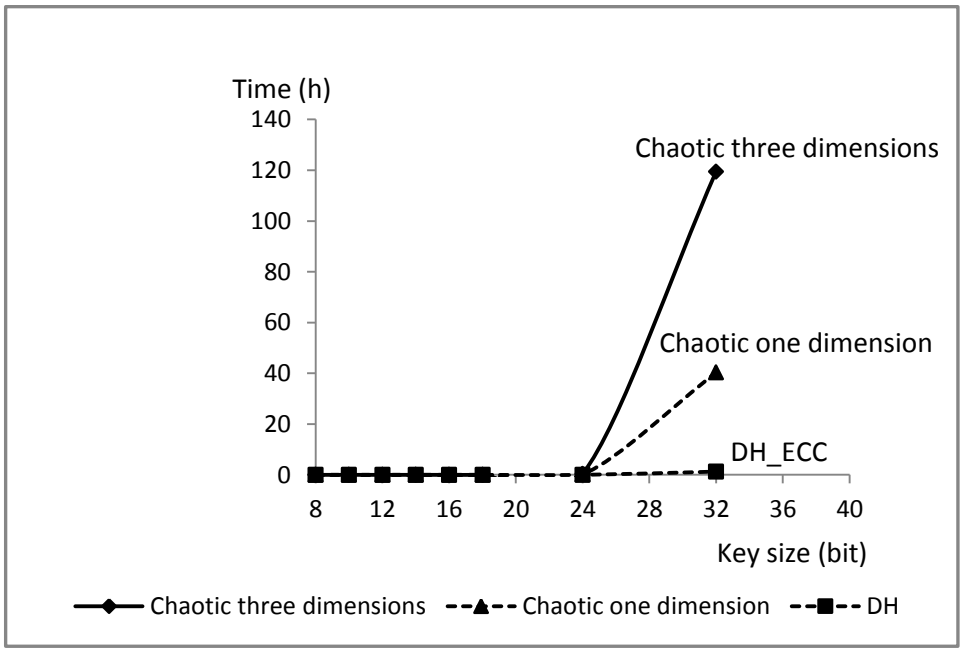



**Fig.10. Brute force attack time for DH_ECC , Chaotic cryptosystems**

c) 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 because the encryptor can choose any number of the iterations.

## 6. Conclusions

This paper presents a chaotic based 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 and sharp privacy for open PGP. Evaluation and comparison with standard mechanism are performed. The results obtained indicate that proposed chaotic public-key cryptosystem enhances both performance and security.

## References

[1] X. Lai, J. L. Massey and S. Murphy, "Markov Ciphers and Differential Cryptanalysis", Advances in Cryptology–EUROCRYPT '91, Lecture Notes in Computer Science, Springer-Verlag, pp. 17-38, 1991.

[2] FIPS PUB 46-3, "Data Encryption Standard (DES)", Federal Information Processing Standards (FIPS), Publications (46-3), National Institute of Standards and Technology, US Department of Commerce, Washington D.C., October 1999.

[3] ANSI X9.52, "Triple Data Encryption Algorithm Modes of Operation", American National Standards Institute, July 29, 1998.

[4] FIPS PUB197, "Advanced Encryption Standard (AES)", Federal Information Processing Standard (FIPS), Publication 197, National Institute of Standards and Technology, US Department of Commerce, Washington D.C., November 26, 2001.

[5] RSA Laboratories, "PKCS #1 v2.1: RSA Cryptography Standard", RSA Security Inc, June 2002.

[6] W. Diffie and M. Hellman, "New directions in cryptography", IEEE Trans. Inform. Theory, vol. IT-22, pp. 472-492, 1976.

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

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

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

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

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

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

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

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

[15] E. Klein, R. Mislovaty, I. Kanter, and W. Kinzel, “Public channel cryptography using chaos synchronization”, Phys. Rev. E., vol.72, 2005.
[16] R. Bose, “Novel public key encryption technique based on multiple chaotic systems”, Phys. Rev. Lett., vol. 26, 2005.
[17] 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.
[18] L. Zhang, “Cryptanalysis of the public key encryption based on multiple chaotic systems”, Chaos, Solitons and Fractals, vol.37, pp. 669-674, 2008.
[19] 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.
[20] Gregory Vert and Manaf Alfize, “An Enhanced Pretty Good Privacy (EPGP) System with Mutual Non-Repudiation”, Security and Management CSREA Press, pp. 364-370, 2006.
[21] Behrouz A. Forouzan, “Cryptography and Network Security”, McGraw-Hill Higher Education, 2008.
[22] William Stallings, “Cryptography and Network Security Principles and Practice Fifth Edition”, Prentice Hall, 2010.
[23] J. C. Sprott, “High-Dimensional Dynamics in the Delayed Hénon Map”, Electronic Journal of Theoretical Physics 3, pp. 19-35, 2006.
[24] 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.
[25] 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.
[26] O. E. Rossler, “An Equation for Continuous Chaos”, Phys. Lett. A, vol. 57, no. 5, pp. 397-398, 1976.
[27] C. Sparrow, “The Lorenz Equations in Chaos”, V. Holden. Princeton. University Press, Princeton, 1986.
[28] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, “Determining Lyapunov exponents from a time series”, Physica D 16, 1985.
[29] W. Meier, “On the Security of the IDEA Block Cipher", Advances in Cryptology-EUROCRYPT '93, Lecture Notes in Computer Science, Springer-Verlag, 1994.
[30] K. Jallad, J. Katz, and B. Schneier, “Implementation of Chosen-Ciphertext Attacks against PGP and GnuPG”, In Proceedings of the 5th International Conference on Information Security, pp. 90-101, 2002.
[31] J. Katz and B. Schneier, “A Chosen Ciphertext Attack against Several E-Mail Encryption Protocols”, InProceedings of the 9th USENIX Security Symposium, pp. 241-246, 2000.
[32] E. Biham, “Cryptanalysis of the chaotic-map cryptosystem”, Proc. Eurocrypt ’91, pp. 532-534, 1991.