A Cryptographic Algorithm based on Bilinear Transformation

Phani krishna kishore M [1] and Venugopal IVS [2]

**[1] Professor, GVP College of Engineering, Department of IT
Visakhapatnam, Andhra Pradesh, 530048, India**

**[2] Asst. Professor, GVP College of Engineering, Department of IT
Visakhapatnam, Andhra Pradesh, 530048, India**

Abstract

The technique of producing ciphers via transformations from one domain to another domain has been widely studied. In this paper a new cryptographic algorithm is presented based on complex bilinear transformations, the strength and performance of the algorithm is analyzed.

Keywords: Cryptography, Complex Bilinear Transformations, Encryption, Decryption.

1. Introduction

Security has become an important aspect in the digital world with the advent of technological advancements. Though the art of designing new methods and algorithms for secure transmission of data dates back to centuries it emerged as a separate branch of technology during the last few years. Several researchers devised methods for encryption and decryption based on mathematical functions.

The Conventional cryptography uses symmetric key algorithms in which a common key is shared between the sender, receiver and operates block wise. Most of the modern symmetric ciphers operate at bit level and largely designed around fiestel structure. DES, AES, BLOWFISH, IDEA, TWO FISH are some of the popular algorithms. The paradigm shift brought by the RSA algorithm towards the public key cryptosystem has changed the scenario of security systems.

For several years the rich theory of numbers has dominated this domain. For the past few years several algorithms were developed using algebraic theory and other fields of mathematics.

However not much work is reported on methods using complex arithmetic. Recently Dimitrov et al, [4] in their work discussed some algorithms for multi exponentiations based on complex arithmetic.

Elsayed Mohammad et al described the elliptic curve cryptography over Gaussian integers [5].

Mohammad Ahmad Alia et al presented a new digital signature scheme based on Mandebrot and Julia fracal sets [8]. In this paper a new symmetric key algorithm that operates block wise based on complex bilinear transformation is proposed.

2. Bilinear Transformation

A mapping f: C → C, (C denotes complex plane) of the form w= f (z) = $\frac{az+b}{cz+d}$ (ad-bc≠0) where a, b, c and d ϵ C,

are constants and z=x+iy, w=u+iv be complex variables in different planes, is called a bilinear transformation from C to C. In fact several transformations can be considered from C to C, however this transformation is a one-to-one mapping from C to C. A bilinear mapping is called conformal if it preserves the angles and orientations between the curves under the transformation. A bilinear transformation is actually a combination of translation, dilation, rotation and inversion.

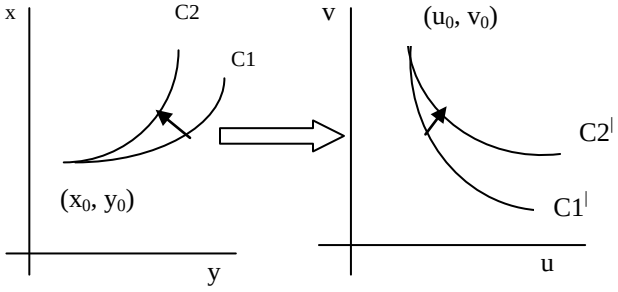

Fig. 1 Bilinear Transformation from one curve to another.

These transformations are widely studied in the theory of complex variables and got many applications in several branches of engineering.

3. Proposed System

In this proposed system data is encoded onto a complex plane and then using a bilinear transformation the data points are transformed into a another set of points in another complex plane producing the cipher text. The transformation being reversible the original data can be obtained using inverse transformation. For the application of the transformation a, b, c and d (constants) act as key. Several methods can be adopted to exchange these keys. However a new system is developed to identify the a, b, c and d based on a preshared information, which actually prevents from transmission of a, b, c and d. Initially sender and receiver will agree on a common text message. A hash code is developed from the text message from a randomly chosen part of the text which is used in the construction of the key.

A common hash table is also shared that consists of information to retrieve information regarding computation of key.

The encryption algorithm makes use of pseudorandom numbers to select the index for the starting point of shared text and to select indicator from hash table to generate the key. The encryption algorithm generates a random number within a specified range that acts as an index for the selection of position of the shared text. The selected text is transformed into 128-bit hash code by MD5.Again another pseudorandom number is generated to identify a splitting criteria from the hash table, the 128 bit hash code is separated into four parts by using the split values obtained from the hash table, that are converted to integers to form a, b, c and d.

The text to be encrypted is processed in blocks of seven characters each. Each block is converted into equal parts of binary and then into decimal x, y to form x+iy(=z) (zeros are appended if necessary).

By using a, b, c and d w= $\frac{az+b}{cz+d}$(=u+iv) is computed.

Then w together with the pseudorandom numbers used for the block are converted into text form that forms the cipher text. A predefined protocol for embedding the pseudorandom numbers can be used. (Eg: 128 bits after first 10 bits or at the end of w etc;)

On receipt of cipher text, receiver separates pseudorandom numbers and u+iv. Receiver computes a, b, c and d by using the pre shared information and pseudo random numbers.

By using z= $\frac{b-wd}{wc-a}$, the receiver gains the text message from x+iy.

4. Algorithms

4.1 Generation of keys(on sender side)

Input: Common text message, hash table.
Output: values of a, b, c and d.
Step 1: Generate pseudorandom number(r_1) ($0<r_1<$length(shared text))
Step 2: Starting at r_1^{th} character the shared text is converted into binary and 128 bit binary code is generated using MD5.
Step 3: Generate pseudorandom number(r_2) ($0<r_2<N$) (N denote the maximum number of splits stored in hash table).
Step 4: Select a particular split values to generate a, b, c and d from hash table using r_2. (Number of bits to compute a, b, c and d respectively).
Step 5: Divide 128 bits obtained in step 2 into four parts and convert the bits into decimal form as per values obtained in step 4.

4.2 Algorithm for encryption

Input: Plain text
Output: Cipher text
Step 1: Split the plain text into block of seven characters each and convert into binary form (zeros are appended to binary form to make it a multiple of 2).
Step 2: Compute x, y by splitting binary form obtained in step 1 into two equal to from z=x+iy.
Step 3: compute w= $\frac{az+b}{cz+d}$ (=u+iv)
Step 4: Convert w, r_1, r_2 into text to from cipher text.
Step 5: Repeat steps 1 to 4 until entire text is converted

4.3 Algorithm to generate keys (on receiver side)

Input: common shared message, hash table, cipher text.
Output : a, b, c and d.
Step 1: From the block of cipher text retrieve r_1, r_2.
Step 2: use r_1 to generate 128 bit binary code from common text.
Step 3: using r_2 find the split for 128 bit code from hash table.
Step 4: compute a, b, c and d.

4. 4 Algorithm for decryption

Input: cipher text
Output: original text
Step 1: Convert each cipher text block into decimal equivalent to from w=u+iv.
Step 2: compute z= $\frac{b-wd}{wc-a}$ (=x+iy)

Step 3: Convert x, y into binary form and hence into text form.
Step 4: Repeat the steps 1 to 3 until entire message is decrypted.

5. Example

Consider the plain text: (Empty space is also considered as a character)
"Cryptography is used to encrypt and decrypt data"
The plain text is separated into blocks of seven characters each and corresponding value of z(=x+iy) are given by
Cryptog = 8899535.0 + 1914855.0i
raphy i= 1.5042438E7 + 1.8763881E7i
s used = 1.5106991E7 + 7959072.0i
to encr= 1.5318278E7 + 1603428.0i
ypt and= 1.597533E7+ 16034328i
decryp= 4257518.0 +8174832.0i
t data =1.5237926E7 + 4010144.0i
The pseudo random number generated from common message is 4. Hence plain text is converted into 128 bit binary from "ptography is used to encrypt and decrypt data"
The 128-bit hash code is
00001110001001011011001001001111111110000000111111100011110011001000101010000110011010110001000110111110101000100010011000100011 0
The split for a, b, c and d for first 21 characters is 32, 28, 24, 44. The decimal equivalents are
a(237351503),b(260111485),c(9523405),d(6.74949524999E+12)
The split for b, c, a and d for next 21 characters is 28, 24, 32, 44. The decimal equivalents are
b(14834468),c(16744700),a(2106675405),d(6.74949524999E+12)
The split for c, a, b and d for remaining characters is 36,40,24,28. The decimal equivalents are
c(3797624063),a(553991901520),b(13460023),d(222579782).
W= $\frac{az+b}{cz+d}$ for each block is
Cryptog=0.005514710810825561 -1.060427362652612E-9i
raphy i=0.005514707075943306 -1.4888801636146341E-9i
s used=0.005514708260135056 -1.2527272333429005E-9i
to encr=0.33331102880924857 +1.790851616570252E-5i
ypt and=0.33329850919102816 +3.4948303171133865E-6i
decryp= 0.3333050105270643 +5.385138305935736E-5i
t data = 0.3333050105270643 +5.385138305935736E-5i
w,r_1,r_2 for each block is converted into text form which is given by
As{qvnescqjx"hq!wrge"um!goas{qv!cof!fdas{qv!f`v`
and is sent to receiver.
The receiver receives cipher text as:
As{qvnescqjx"hq!wrge"um!goas{qv!cof!fdas{qv!f`v`
The receiver converts them into equivalent float equivalents.
Receiver has information regarding ciphers(w) as
0.005514710810825561-1.060427362652612E-9i
0.005514707075943306-.4888801636146341E-9i
0.005514708260135056 -1.2527272333429005E-9i
0.33331102880924857 +1.790851616570252E-5i
0.33329850919102816 +3.4948303171133865E-6i
0.3333050105270643 +5.385138305935736E-5i
0.3333050105270643 +5.385138305935736E-5i
The receiver computes a,b, c and d values from algorithm used to generate the key by decryption algorithm.
Hence receiver computes plain text z= $\frac{b-wd}{wc-a}$ and converts it into string equivalent.
The receiver then gets plain text z as
8899535.0 + 1914855.0i 1.5042438E7 + 1.8763881E7i 1.5106991E7 + 7959072.0i 1.5318278E7 + 1603428.0i 1.597533E7+ 16034328i 4257518.0 +8174832.0i 1.5237926E7 + 4010144.0i
The receiver transfers above into string equivalent which forms the plain text:
"Cryptography is used to encrypt and decrypt data"

6. Cryptanalysis

The strength of algorithm is analyzed in cryptanalysis. Several techniques are proposed in literature namely, linear cryptanalysis, differential cryptanalysis etc. The present method produces cipher text using a mathematical function and operates on blocks of text and hence the following analysis is made, given the algorithm is known.

6.1 Cipher text only

In this case if the attacker knows only the cipher text since there are infinitely many combinations of a, b, c and d it is infeasible to guess a, b, c and d for a text block. Even if a, b, c and d are guessed they are limited to 21 characters.

6.2 Cipher text and corresponding plain text

The knowledge of shared message and cipher text may reveal the 128 bit key, if four blocks of plain text and the corresponding cipher text (z_1, w_1), (z_2, w_2), (z_3, w_3), (z_4, w_4) are known. Then by solving four equations a, b, c and d can be found. However since in this method a, b, c and d

are chosen for every three pairs and hence even if (z_1, w_1), (z_2, w_2), (z_3, w_3) are known then three equations for four unknowns are obtained, its difficult to guess a, b, c and d.
While transmission of cipher text the indices r_1, r_2 are embedded. They can be embedded in variety of ways with a mutual agreement. Even if they are revealed, since the common message and hash table are not known they are not useful to attacker.

6.3 Cipher text, corresponding plain text and key

While decrypting the cipher text even if the attacker grabs the 128 bit key used to generate a, b, c and d there are $^{127}c_4$ possible ways of splitting the 128 bits into blocks and since again a, b, c and d can be arranged in 4! Ways altogether $^{127}c_4$ * 4!= 1488186000 possible ways to split 128 bits to generate a, b, c and d and this has to be carried out for every 21 characters of text.
If a, b, c and d values are taken as complex numbers then if the attacker grabs the 128 bit key used to generate a, b, c, and d there are $^{127}c_8$ possible ways of splitting the 128 bits into blocks and since again complex numbers a, b, c and d(4 real parts and 4 imaginary parts) can be arranged in 8! Ways altogether
$^{127}c_8$ * 8!= 2728665444597882048000 ($>2^{71}$) possible ways to split 128 bits to generate a, b, c and d and this has to be carried out for every 21 characters of text, which further increases the complexity and security of the system. Again, 128 bits in hash table can be rearranged in 2^{128} ways, which correspondingly changes values of a, b, c and d. Hence total number of combinations now become $2^{128} * 2^{71} = 2^{199}$.
The time that is required for computation of w= $\frac{az+b}{cz+d}$ is 0.12 milliseconds. Hence 2^{199} combinations would require 2^{199} * 0.12 = 9.642e+58 milliseconds = 3.056e+48 years.
If the attacker is in possession of hash table and cipher text then has to try 2^{199} combinations by brute force which takes approximately 3.056e + 48 years run on an Intel Pentium-IV 2.4 Ghz Processor.
Even if an malicious user requests to communicate with a valid user exchanges the secret message and hash table, by using different texts and different hash tables for different users this kind of attack can be avoided.
Most of block ciphers with symmetric keys operate at bit level and if once the keys are broken entire message is revealed. In this case since each pair of parameters that are used to generate keys are useful for only 21 characters and hence this method is much stronger.

7. Experimental Results

This method based on mathematical operations, the performance of the method is compared with the much popular public key algorithm RSA.

The time taken for encryption and decryption are observed for different file sizes.

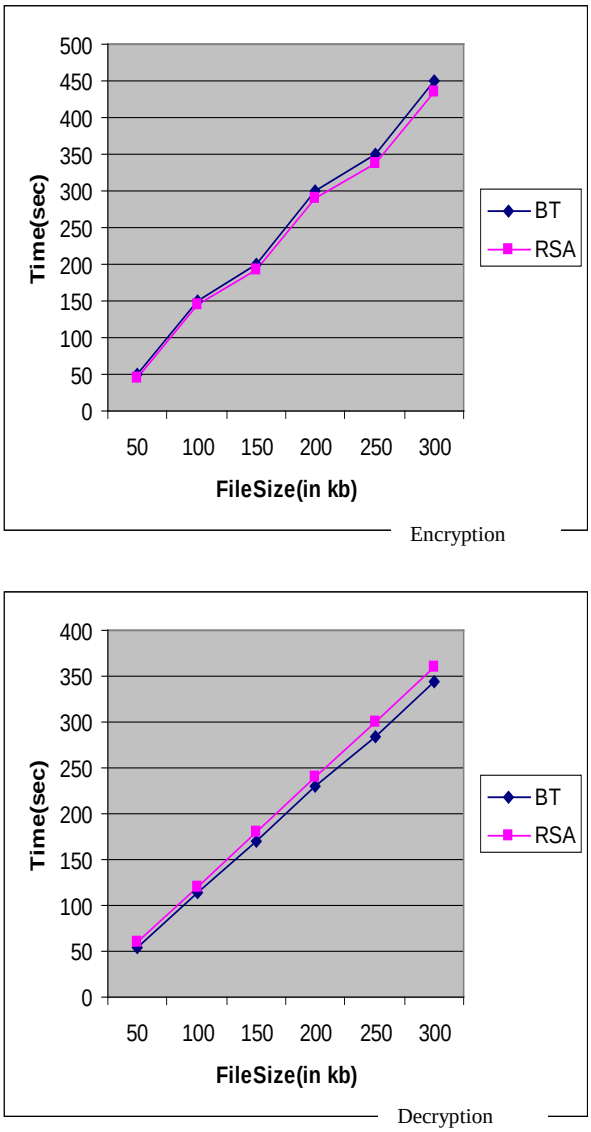

Fig. 2 Comparison between Bilinear Transformation and RSA Algorithms

It is observed that the proposed method takes almost same time to encrypt or decrypt as file size increases. Since the two parameters are embedded into cipher text for every 21 characters the size of the encrypted file will increase in size marginally, however by using efficient data structures like linked lists, each block may contain much larger plain text there by making the increase of encrypted file negligible. The data type 'double' is used to store the values of x, y which restricts the block size to be seven characters. However block size can be increased by using more efficient data types.

8. Usage of the method

This method can be used in different ways. This method can be used in network security applications. Apart from the direct application, the theme can be used in conjunction with any method that involves encoding the text onto a curve or a two dimensional plane. First in case of Elliptic curve cryptography the plain text is encoded onto an elliptic curve. The key used in elliptic curve cryptography can be used to generate the a, b, c and d in bilinear transformation and thereby bilinear transformation can be applied on the encoded points of the elliptic curve.
The images under bilinear transformation form the cipher text thereby providing a double encryption with the same key and some additional preshared information regarding the breakup of key into a, b, c and d.
Secondly, once the text is encoded onto the complex plane, bilinear transformations can be repeatedly applied on the points possibly with different sets of values of a, b, c and d in each round there by increasing the security.

9. Conclusion

In this paper a new cryptographic algorithm has been proposed based on the concept of bilinear transformation from one complex plane to another plane, with additional features the new system is observed to provide more security when compared to existing block ciphers and performance is compared with popular public key algorithm RSA.
The work can be extended to several other complex transformations which are great cryptographic potential.

References

[1] Bruce Schneier, "cryptography engineering", John Wiley & Sons,March15,2010 ISBN: 9780470474242
[2] Christophe De Canniere, Alex Biryukov, and Bart Preneel, "*An Introduction to Block Cipher Cryptanalysis*", proceedings of the IEEE, VOL. 94, NO. 2, December 2008,pp.1274-1281
[3] Chris Karlof David Wagner, "*Hidden Markov Model Cryptanalysis*", Report No. UCB//CSD-03-1244 Computer Science Division (EECS) University of California Berkeley, California 94720,VOL 155,pp.243-249,2009
[4] Dimitrov VS, G.A.Jullien and W.C.Miller VLSI Research Group, University of Windsor, Ontario Canada N9B 3B4,"*Algorithms for multi exponentiations based on complex arithmetic*",VOL.211. pp.208-215.,2009
[5] Elsayed Mohammada and Hassan Elkamchounchi, Alexandria university, Egypt, "Elliptic curve cryptography over Gaussian integers ", International Journal of Computer Science and Network Security, VOL.9 No.1, January 2009
[6] Majithia sachin and Dinesh kumar, "Implementation and Analysis of AES, DES and Triple DES", International Journal of Computer Science and Network Security, VOL.10 No.1, January 2010,pp.298-303
[7]Mircea Andrasiu university Wales – Romania, "StatisticalEvaluationofcryptographic algorithms",IEEE,vol.6,NO.3,January-2009,pp.255-261
[8] Mohammad Ahmad Alia and Azman Bin Samsudin, School of Computer Sciences, Universiti Sains Malaysia, "A New Digital Signature Scheme Based on Mandelbrot and Julia Fractal Sets", American Journal of Applied Sciences 4 (11): 848-856, 2007
[9]RSA Laboratories, http://www.rsa.com/rsalabs/
[10] Song Y Yan, "*Computability, Learnability and Breakability in Cryptanalysis*", volume 45,CIMCA 2008, IAWTIC 2008, and ISE 2008.

Dr.M.Phani Krishna Kishore, is working as a Professor in the department of Information Technology, Gayatri Vidya Parishad College of Engineering,Visakhapatnam. He obtained his Ph.D in 2006, M.Phil in 1998, M.Sc(Mathematics) in 1995, from Andhra University.

Mr.I.V.S.VENUGOPAL, is working as Asst. Professor in the department of Information Technology, Gayatri Vidya Parishad College of Engineering,Visakhapatnam. He obtained his M.Tech in 2010, B.Tech in 2006.