International Journal of Mathematical Archive-7(9), 2016, 43-46
Available online through www.ijma.info ISSN 2229 – 5046

A PUBLIC KEY TRANSFERRING ALGORITHM USING METRIC SPACES

ASHA RANI[1], KUMARI JYOTI*[1], NAVEEN KUMAR ANTIL[2]

[1]Department of Mathematics, SRM University, Haryana, Sonepat- 131001, India.
[2]CITD, Rakuten India, Bangalore- 560054, India.

(Received On: 25-08-16; Revised & Accepted On: 15-09-16)

ABSTRACT

In this paper, we extend the algorithm for the public key transferring presented in the paper "A New Approach To Public Key Transferring Algorithm Scheme Using Metric Space" to a more applicable approach. The algorithm presented in this paper provides a highly secured channel for transferring of data. Moreover, php programming is used to apply the algorithm for practical examples.

1. INTRODUCTION

With the advancement of technology and computer science, to maintain secrecy is a great challenge. But for the armies and other secret agencies this is something of critical need. This need has given birth to the techniques called public key transferring. RSA, Rabin, AlGamal, McEliece, Knapsack, and Probabilistic public key transferring are the common public key algorithms used in practice[2]. But the algorithms present in the literature are all based on the prime numbers and the concepts of number theory. The algorithm presented in [1]is based on the concept of metric spaces. But it has certain drawbacks when applied in the real time programming. Hence we extend the algorithm in a way that it becomes very handy in programing.

In this paper, we have considered the Euclidean metric or 2- metric. In Euclidean metric a hyperball is same as a circle on a plane.

When we seek the security it means there are the factors which need to steal the given information. So, in order to gain the access they can always try to guess the secret keys. For this reason, it is always safe to change the keys frequently. The section of the paper, which introduces the algorithm, we also introduce an algorithm to make changes in the keys by using the same coding when meeting personally is not feasible.

2. PRELIMINARIES

Definition 2.1[3]: Let X be a set and $d: X^2 \to R$ a function with the following properties:

(i) $d(x,y) \geq 0$ for all $x, y \in X$.
(ii) $d(x,y) = 0$ if and only if $x = y$.
(iii) $d(x,y) = d(y,x)$ for all $x, y \in X$.
(iv) $d(x,y) + d(y,z) \geq d(x,z)$ for all $x, y, z \in X$.
Then, we say that d is a metric on X and that (X, d) is a metric space.

Examples 2.2[4]:

1.) The prototype: The line R with its usual distance $d(x,y) = |x - y|$.
2.) The plane R^2 with the "usual distance" (measured using Pythagoras's theorem): $d\big((x_1,y_1),(x_2,y_2)\big) = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$. This is sometimes called the 2- metric d_2.
3.) The metric on the complex numbers $\mathbb{C}$ interpreted as the Argand plane. In this case the formula for the metric is now: $d(z,w) = |z - w|$, where the $|.|$ in the formula represent the modulus of the complex number rather than the absolute value of a real number.
4.) The plane with the taxi cab metric $d\big((x_1,y_1),(x_2,y_2)\big) = |x_1 - x_2| + |y_1 - y_2|$. This is often called the 1- metric d_1.
5.) The plane with the supremum or maximum metric $d\big((x_1,y_1),(x_2,y_2)\big) = max(|x_1 - x_2|, |y_1 - y_2|)$. It is often called the infinity metric d_∞.

Corresponding Author: Kumari Jyoti[1]*

Definition 2.3[5]: In Euclidean *n*-space, an (open) *n*-ball of radius r and center x is the set of all points of distance < r from x. A closed n-ball of radius r is the set of all points of distance ≤ r away from x.

Definition 2.4[5]: A hyperball (hypersphere) is the set of all points of distance = r away from x.

Remark 2.5[5]: In Euclidean *n*-space, every ball is the interior of a hypersphere (a **hyperball**), that is a bounded interval when *n* = 1, the interior of a circle (a **disk**) when *n* = 2, and the interior of a sphere when *n* = 3.

Result 2.6[6]: If we take three circles centred at $(a_1, b_1), (a_2, b_2), (a_3, b_3)$ with radii d_1, d_2, d_3, respectively.

The intersection of these three circles can be discussed in three cases:

(i) No intersection
(ii) One point intersection
(iii) Two point intersection

Theorem 2.7[1]: If three circles intersect in two common points then the centres of the three circles are collinear.

Corollary 2.8[1]**:** If the centres of three intersecting circles are not collinear then the circles intersect at a unique point.

Remark 2.9[1]: If we take distances d_1, d_2 and d_3 of a fixed point X from three points A, B and C respectively. We draw three circles with centres A, B and C and $d_1, d_2$and d_3 as radii. Then, the point X must lie on all of the three circles.

Example 2.10[1]: Let us take the three non collinear points $A(0,1), B(1,0)$ and $C(1,1)$ and an arbitrary point $X(3,4)$. Consider the Euclidean metric

$$d\big((x_1, y_1), (x_2, y_2)\big) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Now, $d_1 = d\big((3,4), (1,0)\big) = \sqrt{20} = 4\sqrt{5}$, $d_2 = d\big((3,4), (0,1)\big) = \sqrt{18} = 3\sqrt{2}$, and $d_3 = d\big((3,4), (1,1)\big) = \sqrt{13}$. Then the intersection of the circles $C_1\left((1,0), 4\sqrt{5}\right)$, $C_2\left((0,1), 3\sqrt{2}\right)$ and $C_3\left((1,1), \sqrt{13}\right)$ will be $(3,4)$.

3. BASIC ENCRYPTION STRATEGY

To first encrypt the message as an integer we use ASCII algorithm. We first bifurcate the message into parts by keeping alphabets in odd places in one place and the alphabets in even places in another. For example if the message is "Gauss!" then it will be bifurcated as "Gus" and "as!". Then we will encrypt the two strings according to the ASCII scheme of encryption. That is the above strings will be encrypted as "071117115" and "097115033", respectively. Now the required point of the message in the space will be $(071117115{,}097115033)$.

The corresponding decryption will be on the same lines as in regular ASCII scheme. That is both the numerical strings are decoded by first grouping the digits in the pair of three from right to left. That is $\big(\overline{071}\,\overline{117}\,\overline{115},\ \overline{097}\,\overline{115}\,\overline{126}\,\overline{033}\,\big)$. Then decoding as per ASCII scheme we get "Gus" and "as!" And hence the message can be rearranged as "Gauss!"

We propose the following algorithms:

3.1. Algorithm for encryption of the message to be sent:

1.) Bifurcate the message in two parts by collecting the alphabets in odd places in one part and the one's in even places in one part.
2.) Encrypt the two parts by ASCII scheme.
3.) Mark the encryption as a point in R^2 as $X(x_1, x_2)$.
4.) Find the square of the distances of X from the keys A, B and C as d_1^2, d_2^2 and d_3^2.
5.) The distances d_1^2, d_2^2 and d_3^2 are the required message coding.

3.2. Algorithm for decryption of the message recieved:

1.) Solve the three quadratic equations of circles to get unique $X = (x, y)$.
2.) Decrypt the point X using ASCII scheme.
3.) Arrange the two strings keeping the alphabets in first coordinate in odd places and in second in even places.
4.) The message so retrieved is the required message.

3.3. Algorithm to send the changes in the keys:

1.) Choose the new keys A_1, B_1 and C_1.

2.) Find the distances of A_1 from A, B and C as d_{11}^2, d_{12}^2 and d_{13}^2.

3.) Similarly, the find the distances of B_1 from A, B and C as $d_{21}^2, d_{22}^2,$ and d_{23}^2 and of C_1 from A, B and C as d_{31}^2, d_{32}^2 and d_{33}^2.

4.) Form the matrix: $\begin{bmatrix} d_{11}^2 & d_{12}^2 & d_{13}^2 \\ d_{21}^2 & d_{22}^2 & d_{23}^2 \\ d_{31}^2 & d_{32}^2 & d_{33}^2 \end{bmatrix}$.

3.4. Algorithm to trace the required changes in the keys:

1.) Solve the three quadratic equations of circles whose squared radii are d_{11}^2, d_{12}^2 and d_{13}^2 to find the first key $X_1 = (x_1, y_1)$.

2.) Similarly, Solve the three quadratic equations of circles whose squared radii are d_{21}^2, d_{22}^2 and d_{23}^2 to get the second key $X_2 = (x_2, y_2)$.

3.) Finally, Solve the three quadratic equations of circles whose squared radii are d_{31}^2, d_{32}^2 and d_{33}^2 $X_3 = (x_3, y_3)$.

4.) The intersection points so found are the new keys.

Example 3.5: Let the three points to be used for sending codes be (0,1), (1,0) and (1,1). Now the distances d_1^2, d_2^2 and d_3^2 can be calculated as follows:

Let the encrypted point be $X = (x, y)$. Then

$$d_1^2 = x^2 + (y-1)^2$$

$$d_2^2 = (x-1)^2 + y^2$$

$$d_3^2 = (x-1)^2 + (y-1)^2$$

Now the next task is to solve this equation for x and y, when d_1^2, d_2^2 and d_3^2 are known. Since, we know that the intersection of these equations is unique, we can solve these as follows:

$$x = \frac{d_1^2 - d_3^2 + 1}{2}$$

$$y = \frac{d_2^2 - d_3^2 + 1}{2}$$

For example, if we take our message to be "Gauss!", then the encrypted point will be, $(071117115, 097115033)$. Then,
c_1 = 14488973486284249, c_2 = 14488973538280085 and c_3 = 14488973344050020, x = 71117115 and y = 97115033, where c_1, c_2 and c_3 are the distances d_1^2, d_2^2 and d_3^2.

So, the encoded message is: 14488973486284249, 14488973538280085, 14488973344050020, which can be decoded as: 71117115, 97115033.

Similarly, if the message is "genius" then it will be coded as $(103110117, 101105115)$. Then, c_1=20853940300696681, c_2 = 20853940304706685 and c_3 = 20853940098486452, x = 101105115 and y = 103110117.

So the encoded message is: 20853940300696681, 20853940304706685, 20853940098486452, which can be decoded as: 101105115, 103110117.

Example 3.6: Let, the keys which are in practice at the moment are $(0,1), (1,0)$ and $(1,1)$. If the new keys to be sent are $(20,30), (70,80)$ and $(20,80)$. Then, c_{11} = 1241, c_{12} = 1261 and c_{13} = 1202 c_{21} = 11141, c_{22} = 11161 and c_{23} = 11002 31 = 6641, c_{32} = 6761 and c_{33} = 6602 x_1 = 20 and y_1 = 30 x_2 = 70 and y_2 = 80 x_3 = 20 and y_3 = 80.

So, the encoded message will be:

$$\begin{bmatrix} 1241 & 1261 & 1202 \\ 11141 & 11161 & 11002 \\ 6641 & 6761 & 6602 \end{bmatrix}$$

which can be decoded as:

$$(20,30), (70,80), (20,70).$$

4. LEVEL OF SECURITY PROPOSED

The basic idea of any public key transferring algorithm is that it secures the message as long as possible. But the persons, who seek the information illegally, are always in the hope to get the access to this information by one way or the other. The public transferring algorithm have their basis on the fact that the message can be read only if one knows the secret keys to access the message. Now, if somehow one can get access to the secret keys, then the concerned person can easily read the information.

Every public key transferring algorithm can be analysed for the security by the fact that how easy or complicated it is to break the coding or to find the keys. Now, the algorithms present in the literature are all based on the large prime numbers. The key is generally a 200 digit number which is a multiplication of two 100 digit prime numbers. So, if somehow one has the access to the encoded message then the person has to guess the key to actually decode the message. But the whole arithmetic revolves around the fact that the key is a composite number which is the multiplication of two large primes of approximately same size. In the days of technology the hackers have advanced techniques of guessing the keys and breaking the codes. In this regard we propose a new highly secure version of sending the encoded message. Because the message codes are sent as the distances from three distinct points which are the keys of our system. If somehow one has the access to the secured channel and the person finds out the encoded message then to actually reach the message one has to know the three points from which the distances has been taken. Now, all the three points can be chosen in such a way so that there is no bar on what is the size of the numbers used in different keys. Moreover, the points can be simple 2- 3 digit numbers and still the security will be intact, which is not possible when we take small prime numbers in the present algorithms. Hence this algorithm is also very simple if we concern to the programming.

In this paper the algorithm presented is more secure, simple to program, and fast in processing. It does not need super computers to program but even a simple personal or office computer is sufficient.

CONCLUSION

The algorithms currently practised in security purposes to send a secret message are all based on the prime numbers. In this paper, we propose the coding to be done using metric distances. This leaves the hacker to rely on pure guessing to actually decode the message. The message can be properly decoded only when the person has the access to the private keys.

SCOPE OF THE STUDY

In programming, the proposed algorithm works very well when the length of character string to be encoded is less than or equal to six. However, the message length can be greater than six in practice.

REFERENCES

1. Asha Rani, Kumari Jyoti "A new approach to public key transferring algorithm scheme using metric spaces", IOSR Journal of Mathematics, Volume 12, Issue 1, 2016, pp-04-06.
2. A. J. Menezes, P. C. van Oorschot, S. A. Vanstone "Handbook of Applied Cryptography", CRC Press ISBN: 0-8493-8523-7, 1996, 816 pages.
3. S. Punnusamy, "Foundations of Functional Analysis", CRC Press, 2002,457 pages.
4. http://www-history.mcs.st-and.ac.uk/~john/MT4522/Lectures/L5.html.
5. https://en.wikipedia.org/wiki/Ball_(mathematics).
6. C. I. Delman, G. C. Galperin, "A Tale of Three Circles," Mathematics Magazine, Santa Clara University, Vol. 76, no. 1, 2003.

Source of support: Nil, Conflict of interest: None Declared