International Journal of Innovative Research in Information Security (IJIRIS) ISSN: 2349-7017(O)
Issue 2, Volume 2 (February 2015) ISSN: 2349-7009(P)
www.ijiris.com

COORDINATE BASED ENCRYPTION PROTOCOL

Shamindra Parui* Darunjeet Bag
Department of ComputerScience Department of Computer Science
St. Xavier's College (Autonomous), Kolkata Panskura Banamali College, Vidyasagar University

***Abstract*— There are several security mechanisms available today. Most of the communication systems, web-sites use SSL/TLS services. So, communication channel is protected. Although unauthorized users may access any confidential data from a distance. Neither do they need to be present at the receiving end nor do they attack the communication channel. Our aim is to restrict the unauthorized access to confidential data based on coordinate system. To properly decrypt the information, a receiver must be present at a particular coordinate. If an unauthorized person tries to decrypt the confidential data, he/she will fail. In this paper we are extending the strength of encryption using four parameters namely; coordinates of sender/receiver, image, key and distance.**

***Keywords*— Cryptography, GPS, Encryption, Hash, Symmetric key, Attack**

I. INTRODUCTION

To communicate with the world people use Internet service. Communication process must be protected by some mechanism. Cryptography helps us to protect our information. Sender/Receiver uses a single key in symmetric-key based systems and uses a pair of keys in asymmetric key based systems. In real world, symmetric and asymmetric key protocols may not be sufficient to secure our information. An attacker may try to snoop into the communication from either side of the sender/receiver's place. Generally an attacker is not physically present at the receiver's end. Since GPS information is accessible easily, this GPS information can act as an individual parameter of cryptographic algorithms. In other words we can say that geo-location restricts the decryption process which must be done on a particular location. RSA algorithm is used with geo-location information[1]. V. Rajeswari, V. Murali, and A.V.S. Anil proposed geo-location and time as the extra parameters to extend the strength of cryptographic method [2]. To minimize the receiver-location's noise while decrypting, a 'Toleration distance' was proposed in 2008 [3]. In this paper we are proposing geo-location, distance between sender and receiver (using Haversine formula) [5], and image as an extra parameters to AES algorithm. With these three parameters we are making a strong key and we are using that key to AES algorithm. Encryption process takes reciever's geo-location, an image, and a key. It also fetches the sender's geo-location from the GPS device and hence encrypts the message (as shown in figure 1). Decryption process requires a receiver to be present at the particular location which was earlier used by the sender (as shown in figure 2). Decryption process also requires sender's geo-location, key, and the image (which acts as an extra parameter). Decryption process fails to decrypt successfully if any of the parameter along with the geo-location is mismatched.

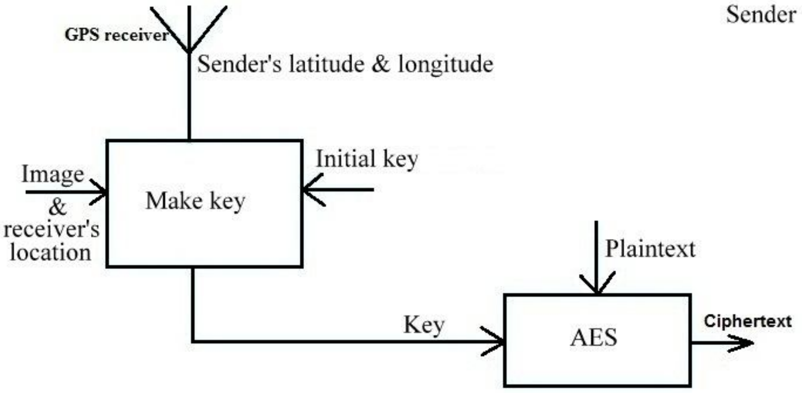


Figure 1: Encryption at sender's end

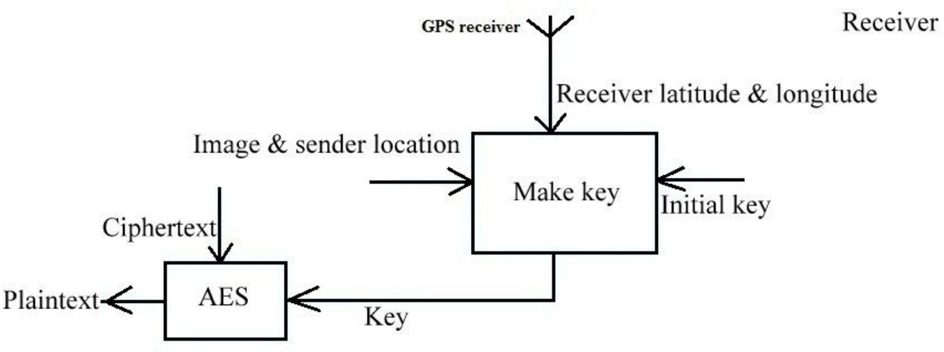


Figure 2: Decryption at receiver’s end

II. COORDINATE SYSTEM

Latitude line is parallal to the equator and longitude line is perpendicular to the equator. Every place on earth can be defined by intersecting point of these two. Lines representing latitude are parallal to each other from 0° (Equator) to 90° north and south. Also, lines represnting longitude are perpendicular to each other from 0° (Prime meridian) to 180° east and west. Hundreds digit in a coordinate represents longitude; tens digit represents a location upto 1000 kilometers and units digit represnts a location upto 111kilometers [4]. Decimal places of a coordinate gives accuracy of a location as follows :

TABLE I

Decimal place	Accuracy upto
First decimal place	11.1 kilometer
Second decimal place	1.1 kilometer
Third decimal place	110 meter
Fourth decimal place	11 meter
Fifth decimal place	1.1 meter
Sixth decimal place	0.11 meter
Seventh decimal place	11 milimeter
Eighth decimal place	1.1 milimeter
Ninth decimal place	110 microns

III. DISTANCE

Haversine formula is used to measure distance between two coordinates over earth’s surface . It calculates the shortest distance between two points on earth [5]. The formula is :

$$a = sin^2(\Delta\emptyset \div 2) + cos\emptyset_1 . cos\emptyset_2 . sin^2(\Delta\lambda \div 2)$$
$$c = 2. atan2(\sqrt{a}, \sqrt{(1 - a)})$$
$$distance = r.c$$

where Ø is latitude, λ is longitude, r is the Earth’s radius (6371 km).

IV. AES

AES (Advance Encryption Standard) is a symmetric key algorithm which is a block cipher. On October 2, 2000 Rijndael was selected as AES. AES has a block length of 128-bits. Three allowable key length in AES are: 128 bits, 192 bits and 256 bits. AES is an iterated cipher. The number of rounds (N) depends on the key length: N = 10 for 128-bit keys, N = 12 for 192- bit keys and N = 14 for 256-bit keys.

V. SHA-1

In Cryptography, SHA-1 is a cryptographic hash function designed by United States National Security Agency and is a U.S. Federal Information Processing Standard published by the United States NIST. SHA-1 is very similar to SHA-0, but alters the original SHA hash specification to correct alleged weaknesses. SHA-1 works with any input message that is less than 2^{64} bits in length. The input of SHA-1 consists of 512 bit blocks. The output of SHA-1 is a message digest which is 160 bits in length. SHA-1 uses 80 rounds. Input which is not conform to integer multiples of 512, bits are padded before any block is inputted to the hash function. It is designed to be computationally infeasible to:

(a) Obtain the original message, given its digest, and
(b) Find two message producing the same digest

VI. PROPOSED WORK

Our algorithm extends the use of coordinate system in encryption-decryption process. At sender's end, algorithm retrieves sender's coordinate from GPS device. Also takes receiver's coordinate, an image and a key along with plaintext. At receiver's end algorithm retrieves receiver's coordinate from GPS device; takes sender's coordinate, key, image and cipher text. Here, image acts as a vector of key. Our algorithm works as three phases. First phase combines i) sender-receiver coordinates, ii) pixel values and iii) distance between sender & receiver. Second phase computes hash value of the result from first phase. And finally third phase encrypts the plaintext with key resulted from second phase. Algorithm works as follows:

1. Retrieve sender's coordinate from GPS device and take receiver's coordinate
2. Take image, key& plaintext
3. Calculate distance (Θ) between sender and receiver using Haversine formula
4. Assign a bit at MSB to represent wheather receiver's latitude and longitude values are negative or positive, to X and Y. A single bit of 1 represents negative number and 0 represents positive number.
5. Convert receiver's coordinate to its corresponding absolute integer value
6. Convert receiver's latitude and longitude to 29 bits binary each and append to X and Y
7. Reverse the receiver's latitude and longitude
8. Repeat for N times, where N is the length of the key :
    I. Retrieve a 24 bits pixel value from image(X,Y) where X = (integer value of receiver latitude Mod image height) & Y = (integer value of receiver longitude Mod image width)
    II. Add extra 6 "0" bits to the right of the 24bit pixel
    III. Calculate X = (receiver's latitude XOR pixel bits) and Y = (receiver's longitude XOR pixel bits)
    IV. Circular Left Shift X and Y by 1 bit
9. Append X and Y and store it to S
10. Convert key and integer value of distance(Θ) to bit representation
11. Append key and distance and store it to J
12. Calculate K = (J OR S)
13. Compute H = SHA1 of K
14. Encrypt the plaintext with H as the key using AES algorithm

At receiver's end, phase 1 and phase 2 remain same; third phase decrypts the cipher text. Decryption works as follows:

1. Retrieve sender's coordinate from GPS device and take receiver's coordinate
2. Retrieve image, key& ciphertext.
3. Calculate distance (Θ) between sender and receiver using Haversine formula
4. Assign a bit at MSB to represent weather receiver's latitude and longitude values are negative or positive, to X and Y. A single bit of 1 represents negative number and 0 represents positive number
5. Convert receiver's coordinate to its corresponding absolute integer value
6. Convert receiver's latitude and longitude to 29 bits binary each and append to X and Y
7. Reverse the receiver's latitude and longitude
8. Repeat for N times, where N is the length of the key :
    I. Retrieve a 24 bits pixel value from image(X,Y) where X = (integer value of receiver latitude Mod image height) & Y = (integer value of receiver longitude Mod image width)
    II. Add extra 6 "0" bits to the right of the 24bit pixel
    III. Calculate X = (receiver's latitude XOR pixel bits) and Y = (receiver's longitude XOR pixel bits)
    IV. Circular Left Shift X and Y by 1 bit
9. Append X and Y and store it to S
10. Convert key and integer value of distance(Θ) to bit representation

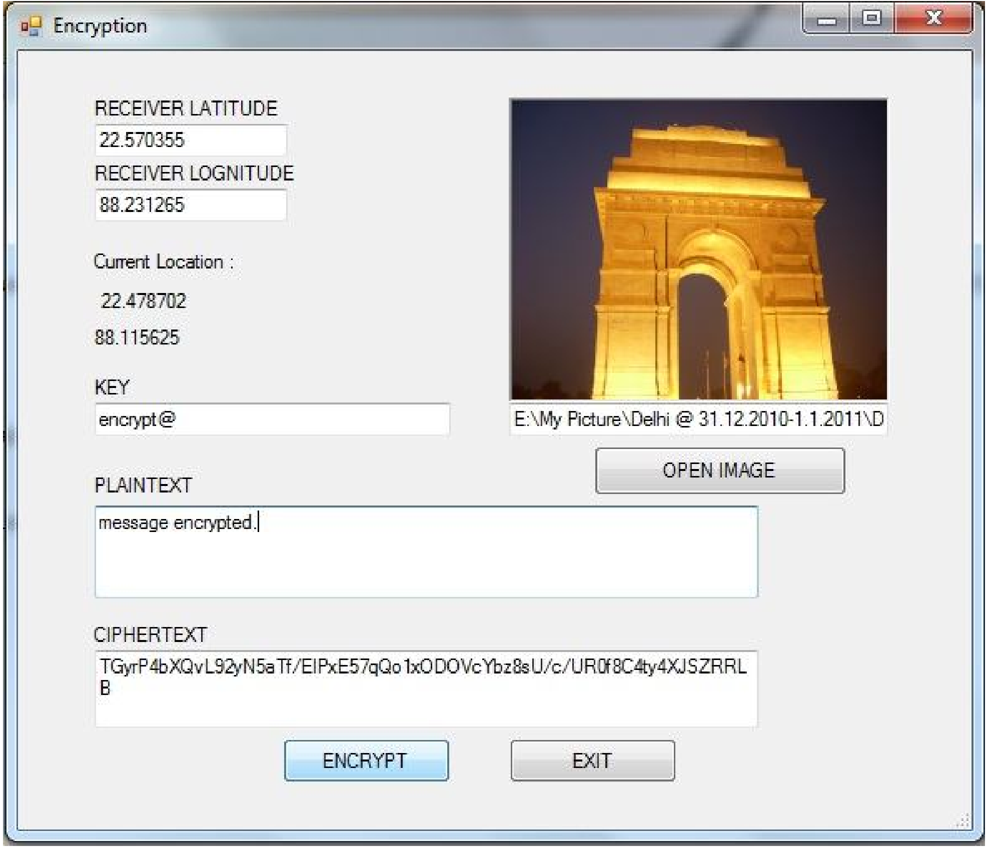
11. Append key and distance and store it to J
12. Calculate K = (J OR S)
13. Compute H = SHA1 of K
14. Decrypt the ciphertext with H as the key using AES algorithm

VII. SECURITY ANALYSIS

Latitude as well as Longitude is represented by a 30 bits binary number. A total of 60 bits is required to represent a coordinate. After performing bit-wise XOR operation twice, hash value is computed to make it a unique key of 128 bits. 2^{128} operations are needed to break this key. Since this protocol uses particular coordinates only and also the sender-receiver distance to decrypt the message, decryption is not possible from any other coordinates.

VIII. EXPERIMENTAL RESULT

Screen-shot of our prototype is as follows:



RECEIVER LATITUDE
22.570355
RECEIVER LOGNITUDE
88.231265

Current Location :
22.478702
88.115625

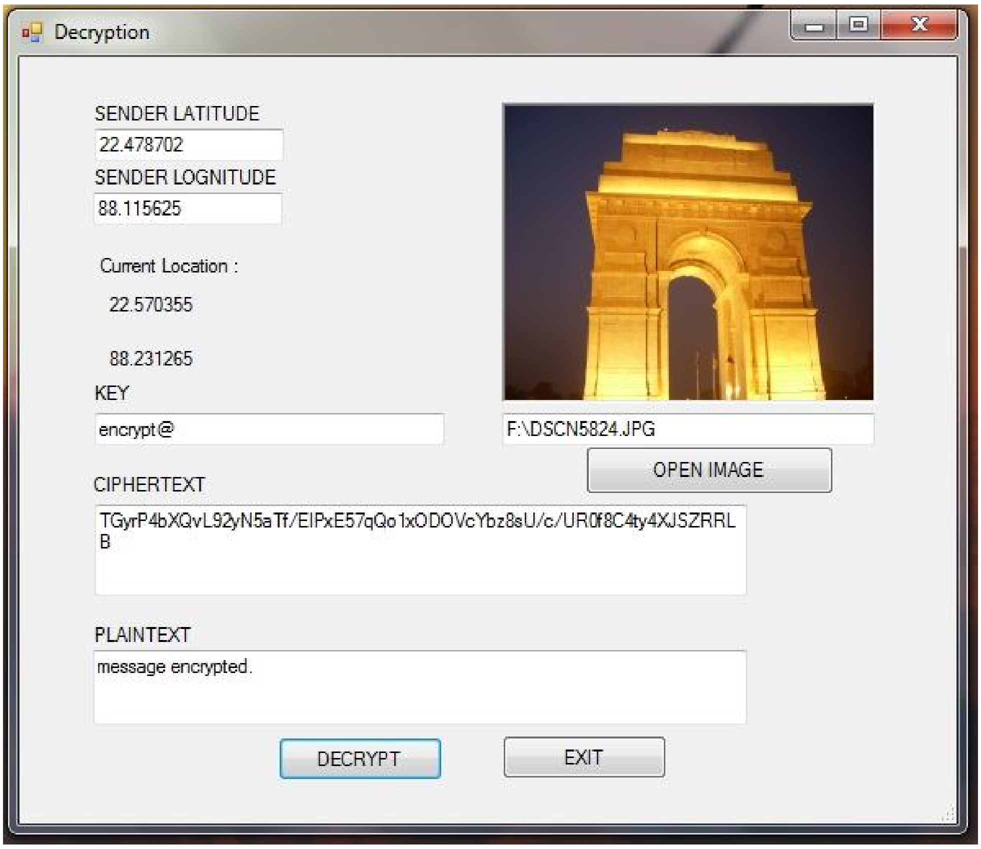
KEY
encrypt@

E:\My Picture\Delhi @ 31.12.2010-1.1.2011\D

PLAINTEXT
message encrypted.

CIPHERTEXT
TGyrP4bXQvL92yN5aTf/EIPxE57qQo1xODOVcYbz8sU/c/UR0f8C4ty4XJSZRRL
B

Figure 3: Screen-shot of encryption



Decryption

SENDER LATITUDE: 22.478702

SENDER LOGNITUDE: 88.115625

Current Location :
22.570355
88.231265

KEY: encrypt@

F:\DSCN5824.JPG

CIPHERTEXT: TGyrP4bXQvL92yN5aTf/EIPxE57qQo1xODOVcYbz8sU/c/UR0f8C4ty4XJSZRRLB

PLAINTEXT: message encrypted.

Figure 4: Screen-shot of decryption

IX. CONCLUSION & FUTURESCOPE

Day by day attackers are trying to apply newer attacks on security systems. With coordinate based encryption mechanism it is possible to stop attackers to decrypt the message from other place. More restriction can be applied on this location based encryption protocol by applying the network delay time. Also zero-knowledge authentication method can be added to provide authentication for both sender and receiver parties.

REFERENCES

[1] Ayesha Khan, "Geo Location Based RSA Encryption Technique", International Journal on Advanced Computer Theory and Engineering, volume 2, issue 2, 2013, ISSN : 2319-2526

[2] V. Rajeswari, V. Murali, A.V.S. Anil, "A Navel Approach to Identify Geo-Encryption with GPS and Different Parameters (Locations And Time)", International Journal of Computer Science and Information Technologies, Volume 3 (4), 2012, ISSN : 0975-9646

[3] Hsien-Chou Liao, Yun-Hsiang Chao, "A new data encryption algorithm based on the location of mobile users", Information Technology Journal, 7 (1), 63-69, 2008, ISSN : 1812-5638

[4] URL : www.gis.stackexchange.com/questions/8650/how-to-measure-the-accuracy-of-latitude-and-longitude

[5] URL : http://www.movable-type.co.uk/scripts/latlong.html

[6] URL : http://en.wikipedia.org/wiki/SHA-1