DOI: http://dx.doi.org/10.26483/ijarcs.v8i7.4208

ISSN No. 0976-5697

Volume 8, No. 7, July – August 2017

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

ENCRYPTION/DECRYPTION USING ELLIPTICAL CURVE CRYPTOGRAPHY

Ansah Jeelani Zargar
Student, Computer Science Engineering,
SSM College of Engineering and Technology,
Kashmir, India

Mehreen Manzoor
Student, Computer Science Engineering,
SSM College of Engineering and Technology,
Kashmir, India,

Taha Mukhtar
Assistant Professor, Computer Science Engineering,
SSM College of Engineering and Technology, Kashmir, India,

***Abstract*:** One of the buzzwords in network security nowadays is the ECC-Elliptical Curve Cryptography. It is one of the best cryptography techniques that provide security to our personal as well as professional data over the network. In our day to day life the need of data sharing has increased exponentially. We like to stay updated with every events occurring around and love to be a "know-it-all". With the upswing of social networking people stay connected all the time. They make their social media profiles and use them to communicate by sharing their information over the network. It has become a necessary part of our life. The need for securing that data has also increased, in order to prevent attacks that may cause unauthorized access to our data, misuse of our data or modification of our data and also to maintain privacy over the network. Thus sharing the data has to be done in such a way such that only the sender and the receiver understands it and no one else on the network does. It is done by using methods of encryption and decryption, wherein we change our data in a way that it becomes senseless to everyone, until it is changed back to its original form in order to make sense out of it. We have many techniques for doing the same. In this paper we will be discussing about two such techniques that encompass not only server and desktop systems, but also large numbers of small devices ranging from PDAs and cell phones to appliances and networked sensors. To the end we will be making a necessary comparison between the techniques explaining why ECC is better than any other cryptographic techniques recent to it.

***Keywords*:** Efficiency, Elliptical Curve Cryptography (ECC), Personal Digital Assistant (PDA),Ron Rivest, Adi Shamir and Leonard Adleman (RSA), Signature Verification.

1. INTRODUCTION

Cryptography [1] has been derived from Greek word "krptos" meaning "hidden or secret" and "graphy" means "writing or study". It is the practice and study of techniques that allow us to make secure communication possible in presence of third parties called adversaries. It constructs and analyses protocols that prevent third parties from reading private messages or confidential data. Confidentiality, data integrity, authentication, and non-repudiation are central to cryptography. Modern cryptography exists in almost every disciplines like mathematics, computer science and electrical engineering. Cryptography involves two methods called encryption and decryption. Encryption changes the plain text to cipher text using encryption algorithms such that no one other than the sender can make sense out of it using a key generated by the algorithm during the encryption process. Decryption is the reverse of encryption that is done on the receiving end. But in order to do it the receiver must have the knowledge of key otherwise he will not be able to make sense out of the received encrypted message. Cryptography is of two types, Symmetric Cryptography and Asymmetric Cryptography. Symmetric encryption (also called as secret-key cryptography) uses a single undisclosed key for both encryption and decryption. We need to keep this key as a secret in the network. However it is very tough to do so in the exposed environments where wireless sensing networks are used to

achieve safety requirements. Several researchers have concentrated on evaluating cryptographical algorithms in

wireless sensing networks and are offering energy efficient ciphers as well. Symmetric key algorithms are much faster computationally than asymmetric algorithms as the encryption process is less intricate. Examples are AES, etc. Asymmetric encryption (also called public-key cryptography) uses two associated keys (public and private) for data encryption and decryption, and takes away the security risk of key allotment. The secretive key is never exposed. A message that is encrypted by using the public key can only be decrypted by applying the same process and using the identical private key. Likewise, a message that is encrypted by using the secretive key can only be decrypted by using the identical public key. Examples are RSA, ECC etc.

2. EXISTING SYSTEM

RSA ALGORITHM:-

It is the widely used encryption algorithm in the world. Originally RSA [2] came from three MIT professors Ron Rivest, Adi Shamir and Leonard Adelman. The acronym RSA has been derived from the initials of their last names.

This algorithm was invented in 1977 and was published the following year in communications of ACM. It was a ground breaking piece of research and the ground breaking about this algorithm was that for the first time we had a system to encrypt with one key and decrypt with another and its inverse was also true. In order to understand RSA we must have a good knowledge about prime numbers, number factorization, Euler's phi function and Euler's totient theorem.

The intuition behind RSA is that suppose we would like to receive a secret message or secret information from other people, so what we need is a key and a decoder. We will keep key as public so that anyone can access it and we will keep decoder as private so that no one other than the recipient of the message (to be decoded) can access it. In this way when people try to send receiver a message they can use its public key and change the secret message by it. In this way the message will makes no sense at all unless, the recipient uses the decoder that will change the message such that it starts making sense again. Thus providing high end security to our data because even if anyone has our received message still without our decoder no one can make use of our data. Thus, key is public key and decoder is the private key that provides security to our messages. Example, we use online banking where we buy a product using its public key which encrypts our information and sends it to server. There on the server they use private key to decode the encrypted data.

RSA ALGORITHM:-

Step 1. Generate two large prime numbers p and q.

Step 2. Calculate the product of these two prime numbers such that the product is represented by n i.e$n = p * q$

Step 3. Calculate the value of phi for n such that $\varphi(n) = (p-1) * (q-1)$it's called the Euler's totient function. The value of phi should satisfy $x^{\varphi} \, mod \, n = 1$and φ must not share a factor with e.

Step 4. Select a random integer e such that it lies between 1 and φ and e, φ are coprime i.e $gcd(e, \varphi) = 1$. Also e should be very small but greater than 2.

Step 5. Calculate inverse of e i.e d such that d lies between 1 and φ and $e * d \, mod \, \varphi = 1$. By using Euclid's Extended Greatest Common Divisor Algorithm.

Step 6. The public key formed will be (n,e) and the private key formed will be (d,p,q) where all the values of d, p, q, and phi are kept secret.

We need to first select two large random prime numbers say p and q with maximum size of 2048 bits each. They must be extremely large. Then in next step we multiply them both and evaluate a new value n that is the product of p and q which intern is extremely large as well, with the size of 4096 bits. Then we use Euler's phi function $\varphi(n) = (p-1) * (q-1)$and calculate the phi value from this function. Afterwards we select a random integer e such that it ranges between 1 and phi. e and phi need to be coprime or relatively prime such that$gcd(e, \varphi) = 1$. In the next step we continue with selecting another random number d which is actually the inverse of e such that d lies between 1 and phi and also satisfies$e * d \, mod \, \varphi = 1$ equation. Thus here we have a public key which is the combination of (n,e) where n is the product of prime numbers called as modulus and e is the random coprime number called as public exponent, including a private key which is a combination of (d,p,q) where d, p, q are the randomly selected values where p and q are prime numbers and d is known as secret exponent.

ENCRYPTION IN RSA:-

Let's assume we have two communication ends one as a sender X and other as a receiver Y. The idea behind RSA is that the receiver publishes it public key over the network which is actually a combination of product of prime numbers (modulus) and random coprime number (public exponent) and keeps its secret key to itself. Now say X has a plain text message and it wants to communicate it with Y. Say the plain text message of X is represented by an integer m such that m lies between 1 and n. For example we have a text message then we convert it into ASCII values such that if their ASCII value exceeds n then we split them into two or more using some operations over it so that the value of m stays less than n. Once the X receives the public key of Y, it is now capable of communicating securely over the network by simply encrypting the m with the public key of the receiver using equation$c = m^e \, mod \, n$. In this equation at first we calculate the value of m raised to power e and then divide it with n which yields us with a remainder c which is our encrypted message or cipher text message. It's this c that is then transferred over the network to the receiver Y.

DECRYPTION IN RSA:-

At the receiving end the receiver is having its private key for decoding the messages meant for it. So once the receiver receives c then it will do the opposite of the encryption process and compute $m = c^d mod \, n$ which will yield the original message. Here cipher message is raised to the power of d and once it's resolved the value is then divided by n whose remainder is evaluated as to be the original message

If we look at both the equations together $(m^e mod \, n)^d mod \, n = m$ then we can clearly see that one is the undoing of another.

WHY WE PREFER RSA?

RSA is called a one way trapdoor function, what it actually means is that there is no way of undoing encryption unless we know the trapdoor. In this case trapdoor is the n. In RSA we have 2 large prime numbers such that a composite number can be computed from them i.e. n=p*q where p and q are prime and n is composite number. But it's difficult to find this trapdoor because from the prime number factorization, and according to the fundamental theorem of arithmetic we know that any number greater than 1 can be written in exactly one way as a product of prime numbers. It is easy to calculate the product of two large prime numbers but it's very hard to take a very large product and factor it to find the two prime numbers that composed it. It's very difficult but not impossible it might take us just 50 years to find out. Thus factors of the n are the actual trapdoor. If know the factors of n only then we can decrypt our encrypted messages.

3. PROPOSED SYSTEM

ELLIPTICAL CURVE CRYPTOGRAPHY

Elliptical curve cryptography [3] [4] is a public key encryption technique which is based on the theory of elliptical curves. This encryption technique uses the properties of elliptic curve in order to generate keys instead of using the traditional methodology of generation of keys using the product of two very large prime numbers. Initially the elliptic curves for cryptography was used in H.W. Lenstra's elliptical curve factoring algorithm. Inspired by this unpredictable use of elliptic curves, the elliptical curve cryptography was proposed by N.Kobiltz and V.Miller independently in 1985. The most important advantage of elliptical curve cryptography is the use of smaller keys providing the same level of security. ECC can provide the same security with 164-bit key that other systems provide with 1024- bit key. It is mostly useful for mobile applications as it has the capability to provide high level security with low computing power and battery resource. ECC is a public key cryptosystem which is used to generate the public key and the private key in order to encrypt and decrypt the data. It is based on the mathematical complexity of solving the elliptic curve discrete logarithm problem which deals with the problem of calculating the number of steps or hops it takes to move from one point to another point on the elliptic curve.

Elliptic curves are the binary curves and are symmetrical over x- axis. These are defined by the function:

$$y^2 = x^3 + ax + b$$

where x and y are the standard variables that define the function while as a and b are the constant coefficients that define the curve .As the values of a and b change, elliptical curve also alters. For elliptical curves, the discriminant $\Delta = 4a^3 + 27b^2$is non zero. The operations used on elliptical curves in cryptography are point addition, point multiplication and point doubling. The important characteristic of elliptic curve is the finite field concept. This means that there is a way to limit the values on the curve. This "max" value established on the x-axis is represented by "p". It is also called "modulo value" for any ECC cryptosystem. This point depicts the finite length upon which the operations can be performed on the curve. In ECC, the modular value depicts the key size for the system. Thus the parameters that fully define the ECC cryptosystem are:

P :- Specification of the finite field

a,b :- Coefficients for defining curve

G :- Generator point on the curve where the operation starts

n :- Order of G

h :- Division of the total points on the curve and he order of G.

STEPSINVOLVED IN ECC ALGORITHM [5] [6]:-

ECC is a public key cryptosystem where every user possesses two keys: public key and private key. Public key is used for encryption and signature verification while as private key is used for decryption and signature generation.

KEY GENERATION:-

It is the most important step in which an algorithm is used to generate both public and private keys. Sender encrypts the message data with the help of receiver's public key and receiver decrypts the data using its private key.

Step 1. The sender selects a random number dA between the range [1 , n-1].This is the private key of the sender.
Step 2. Then the sender generates the public key using the formula PA = dA*G
Step 3. Similarly receiver selects a private key dB and generates its public key PB =dB*G.
Step 4. The sender generates the security key " K= dA*PB" and the receiver also generates the security key " K= dB*PA"

SIGNATURE GENERATION:-

To sign a message m by the sender, it performs the following steps:-

Step 1. It calculates a cryptographic hash function
$$e = HASH(m)$$
Step 2. The sender then selects a random integer k from [1,n-1]
Step 3. The it computes a pair (r,s)
Step 4. r= x1 (mod n) where (x1,y1) =k*G
Step 5. s= k-1(e+ dA*r)
Step 6. This pair (r,s) defines the signature
Step 7. This signature is sent to the receiver.

ENCRYPTION ALGORITHM:-

Suppose sender wants to send a message m to the receiver

Step 1. Let m has any point M on the elliptic curve
Step 2. The sender selects a random number k from [1,n-1]
Step 3. The cipher texts generated will be the pair of points (B1,B2) where
B1= k*G
B2= M + (k*G)

DECRYPTION ALGORITHM:-

To decrypt the cipher text, following steps are performed:-

Step 1. The receiver computes the product of B1 and its private key
Step 2. Then the receiver subtracts this product from the second point B2
M = B2- (dB * B1)
M is the original data sent by the sender

SIGNATURE VERIFICATION:-

To authenticate the sender's signature, the receiver must have the knowledge about sender's public key PA

Step 1. For authentication the receiver needs to verify the pair (r,s) are in the range of [1,n-1]
Step 2. The receiver again then calculates the hash function e as in signature generation
Step 3. Then the receiver calculates w =s-1 mod(n)
Step 4. Then calculate u1= e*w (mod n) and u2 = r*w (mod n)
Step 5. Calculate (x1,y1)= u1*G + u2*PA
Step 6. If x1 = r (mod n), then the signature is valid.

4. COMPARISON BETWEEN RSA AND ECC

The encryption algorithms ECC and RSA both provide same level of security especially at higher levels but the most important advantage of ECC is the use of smaller keys. ECC generates smaller cipher texts and signatures as compared to the linear cryptosystems. The ECC algorithms promises a faster key generation which is contradictory to the RSA algorithm which has a slow key generation algorithm. ECC can provide the same security with 164-bit key that RSA systems provide with 1024- bit key. The RSA algorithm provides slow signing and decryption and thus can be replaced by ECC which is computationally faster in encryption and decryption processes. The ECC signatures can be computed in two stages and are thus more secured than RSA in which signatures are difficult to implement securely. ECC provides excellent protocols for the key exchange while as the two-part key in RSA is very vulnerable to attacks. The elliptical curve cryptography has another plus on its sides as the binary curves are really fast to implement in hardware. Thus paving the way for ECC to become a household name in future.

5. CONCLUSION

Data security is a major issue to be dealt with from a few decades. In this paper we have reasoned the use of cryptographic techniques for securing the data. We have discussedthe two efficient data security algorithms- ECC and RSA. These algorithms help in securing in transit data. Since RSA is a linear cryptographic algorithm and is slow in the encryption and decryption processes, it can put the user's security at risk. Thus ECC is the cryptographic algorithm which provides security and authentication. Authentication to the data is provided with the help of smaller keys. The computational cost as well as the speed of this algorithm is comparatively better. It also makes use of the good exchange protocols giving another mark to the security. In future, we emphasise the use of elliptical curve cryptography for providing high data security in almost all low power devices.

6. REFERENCES

[1]. https://en.wikipedia.org/wiki/Cryptography
[2]. https://en.wikipedia.org/wiki/RSA_(cryptosystem)
[3]. http://www.sciencedirect.com/science/article/pii/S0022314X09000481
[4]. https://en.wikipedia.org/wiki/Elliptic_curve_cryptography
[5]. http://www.academia.edu/download/32863696/Data_Security_in_Cloud_Architecture_Cryptography.pdf
[6]. https://www.researchgate.net/profile/Veerraju_Gampala3/publication/265359508_Data_Security_in_Cloud_Computing_with_Elliptic_Curve_Cryptography/links/587c313308ae9275d4e019eb/Data-Security-in-Cloud-Computing-with-Elliptic-Curve-Cryptography.pdf

7. BIOGRAPHIES

Ansah JeelaniZargar, is pursuing B.E Degree from SSM College of Engineering & Technology in Department of Computer Science Engineering, University of Kashmir, J&K, India. Fields of interest are JAVA and Network Security.

MehreenManzoor, is pursuing B.E Degree from SSM College of Engineering & Technology in Department of Computer Science Engineering, University of Kashmir, J&K, India. . Fields of interest are JAVA and Network Security.

Ms. Taha Mukhtar, is working currently as Assistant Professor at SSM College and Engineering and Technology in Department of Computer Science Engineering, University of Kashmir, J&K, India. Fields of interest are Network Security, Cryptography and Implantable Biosensor Security, Distributed Systems.