SHARING SECRET WITH PUBLIC KEY CRYPTOGRAPHY

Arvind[1*], Meenakshi Agarwal[2]

Abstract

In today's world where Internet has become an inseparable part of our lives, Data security is of utmost importance to us. Every user wants to send his data over a secure channel so that his data can only be accessed by the legitimate user and to make sure that the data has not been tampered while transmission. Public key cryptography is very helpful in achieving a great level of security. In this paper we will talk about different public key cryptography techniques and algorithm. We will analyse RSA (Rivest Shamir and Adleman), Diffie-Hellman, DSA (Digital Signature Algorithm) and ECC (Elliptic curve cryptography). We would also be discussing their strengths and weaknesses. We will analyse their performances. Finally, we will conclude that which algorithm works most efficiently under what circumstances.

Keywords

RSA (Rivest Shamir and Adleman), Diffie-Hellman, DSA (Digital Signature Algorithm), ECC (Elliptic curve cryptography)

1. INTRODUCTION

While communicating through a network everyone wants to transmit his data securely so that no illegitimate user can access the data. Cryptography is used to do a secure data communication through wired and wireless network but what exactly is cryptography? Before knowing about cryptography, it is important to know about cryptology. Cryptology is the study of techniques for ensuring the secrecy and/or authenticity of information. Cryptology classified as cryptography and cryptanalysis. Cryptography is about constructing such techniques; and Cryptanalysis deals with defeating such techniques, to recover information, or forging information that will be accepted as authentic. Cryptography converts readable message into non-readable form. The process of converting from plaintext to ciphertext is known as enciphering or encryption; restoring the plaintext from the ciphertext is deciphering or decryption. Cryptography categorized as Symmetric cryptography and Asymmetric cryptography. In symmetric key cryptography, both parties use the same key. With the help of this key and encryption algorithm, sender encrypts the data; and the receiver decrypts the data by using the same key and the decryption algorithm. In Asymmetric key cryptography, Sender and Receiver uses different keys for encryption and decryption namely PUBLIC and PRIVATE key respectively. Fig. 1.1 shows the basic concept of Public Key Cryptography where the sender encrypts the message using the public key of receiver and sends the encrypted text to the receiver. The receiver receives the enciphered text and decrypt it, using his private key. Note that only the receiver has his private key so only he can decrypt the encrypted text and read the message. Thus in this case, Public key Cryptography has provided confidentiality.

1.*Corresponding Author,Department of Mathematics, Hansraj College,
Email: arvind_ashu12@rediffmail.com
2. Research Scholar, Department of Mathematics, University of Delhi

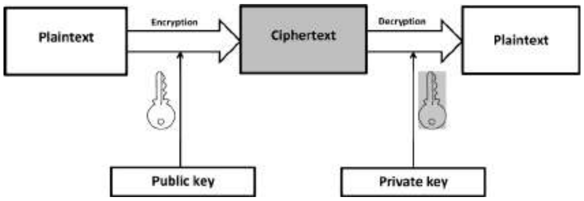


Figure 1.1 Public Key Cryptography Concept

Asymmetric cryptography is also known as Public Key Cryptography. As the name suggests Public key is made public and receiver keeps the private key as a secret. RSA (Rivest Shamir and Adleman), Diffie-Hellman, DSA (Digital Signature Algorithm), ECC (Elliptic curve cryptography) are some of the most frequently used public key cryptography techniques. Among them, the most popular public key cryptography technique is RSA. It is a block cipher i.e. it operates on fixed size of bit groups of plaintext. In this algorithm, plaintext and ciphertext are represented as integer k where, $0 < k < n$ for some n. It offers confidentiality, authenticity, integrity and nondeniability of the data transmitted. It is based on the difficulty of computing the prime factors of product of two large prime numbers. Diffie-Hellman is the simplest public key cryptography technique. It allows two users to exchange a secret key over an insecure channel. It is based on the difficulty of finding discrete logarithms. This technique works fruitfully as long as two parties can mutually authenticate each other. It does not provide authentication. It cannot be used as an encryption/ decryption algorithm. DSA is one of the strongest digital signature technique which is currently being used at many places. It is also based on the difficulty of computing discrete logarithms. It cannot be used to share keys or to perform encryption/decryption. Elliptic curves defined over finite fields are used in developing ECC techniques. It is comparatively a newer member in the family of public key cryptography techniques. It is and has been the “centre of attraction” from past many years for the great level of security it provides, with keys of smaller lengths. We will thoroughly discuss these techniques in this paper. We will also look at the strengths and weaknesses of these techniques with the possible attacks on them.

2. STUDY OF DIFFERENT TECHNIQUES

We will examine the above-mentioned techniques in this paper on the basis of different research paper.

2.1. Rivest Shamir and Adleman (RSA) algorithm

[6] RSA is one of the most popular public key cryptography algorithms. It was introduced by Rivest, Shamir, and Adleman in 1977. It is based on the difficulty of finding large prime factors of integers. To have a better understanding of the need of this algorithm, consider the following problem.

Suppose Ash wishes to communicate with Ben, but they have not previously agreed on a key and they do not wish to send the key in a courier. Thus, all of their messages can be seen by the intruder, Eric. However, they can still communicate with each other in a way that Ben can read all the messages, but Eric cannot.

[4]Let us see how this algorithm works. Two different large prime numbers are chosen by Ben say p and q and then he computes

$n = pq$.

Ben also chooses a number e in such a way that

$gcd((p\text{-}1)(q\text{-}1),e) = 1$.

(n,e) is the public key of Ben. He calculates d such that

$de = 1 \ (\mathrm{mod}(p\text{-}1)(q\text{-}1))$.

(p,q,d) is the private key of Ben. He sends his public key to Ash. Ben does not share his private key with anyone. Ash writes his message in the form of an integer say m. To encrypt the message m, Ash computes

$c = m^e \ (\mathrm{mod}\ n)$

and sends it to Ben. Ben receives the message c. To decrypt c, He calculates

$m = c^d \ (\mathrm{mod}\ n)$.

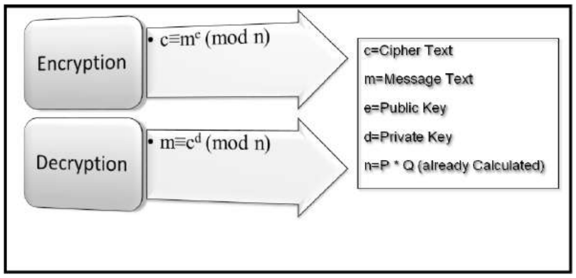


Figure 2.1 The RSA Algorithm

Fig. 2.1 has been taken from the website*https://www.c-sharpcorner.com/UploadFile/75a48f/rsa-algorithm-with-C-Sharp2/*. It gives the brief summary of RSA algorithm. Here, selection of e depends on the selection of prime numbers$p$ and q. If the values of$p$ and q are chosen to be large enough then it would be very difficult to estimate the encryption. [4]For example- Take $p = 7$, $q = 17$. e must not be a factor of $(p-1)$ $(q-1)$

i.e. (7-1)*(17-1) = 6*16 = 96 = 2*2*2*2*3. So, e can take the value 5, 7, 11…

Similarly, we cannot choose a smaller value of d as it can increase the probability of a brute force attack.

Attacks on RSA algorithm

Types of attacks possible on RSA are [6]:

- **Brute Force Attack:** In this attack, the intruder tries every possible key with a hope of getting the right key. We can prevent this attack by using a large key space.
- **Mathematical Attack:** It involves finding the prime factors of n which will help the intruder to find $\ddot{O}(n) = (p-1)\ (q-1)$ and thus enable him to determine $d = e^{-1}$ (mod $\ddot{O}(n)$).

Users may get attracted to the smaller values of$p$ and q as it will speed up the process of encryption and decryption but then it will be a cake walk for the force attacker to find out these numbers.

For example, Take $p = 5$, $q = 3$ and $n = pq = 3*5 = 15$. In this case, It can be clearly seen from the value of n that the value of p and q are 3 and 5. So the two prime factors of$n$ have to be kept as a secret because if someone obtains these values then he can decrypt all the messages.

We can prevent this attack by choosing large key size for n. A key size of around 1024 - 2048 bits seems rational so that an attacker cannot find the prime factors from the value of n but factoring an integer n is not hard in today's computer days so in addition to this, researchers have suggested that with some extra conditions on the value of p and q we can prevent this attack.

1. The length of p and q should differ by a few bit.
2. There should be a large prime factor of $(p-1)$ $(q-1)$.
3. $(p-1)$ and $(q-1)$ should have a small *gcd*.

[2]In the paper presented by Chandra M. Kota et al., It has been shown that if the size of prime factor p and q is less than or equal to the number of one fourth bits present in n then the whole system can be attacked.

- [6]**Timing Attack:** In this attack, an attacker tries to gain information about the plaintext from the computation time of encryption and decryption. We can prevent this attack by making sure that all the computations take same time to be executed.

[4]This algorithm is not viable for wireless communication because the key size is very large. The size of$p$ and q must be no less than 100 digits. It is a very time-consuming algorithm, involves too many calculations and thus not viable for transmitting large amount of data. However, it is easy to implement this algorithm in any software. It is easy to use and upgrade the algorithm. The advantages of RSA lie in its workability and movability.[6] Unmodifiable signatures can be created using RSA algorithm. It is also useful in providing authenticity. It can be used in sending the key of faster private key cryptography securely.

2.2. Diffie-Hellman Algorithm

Two users can use this algorithm to exchange cryptographic keys securely over an insecure channel. RSA is one solution to this problem. Diffie-Hellman algorithm is another solution of this problem. Consider the following situation :

[4]Suppose Ash and Ben wants to share a secret key of symmetric cipher but the only way to share the key is through an insecure channel. All of their message sent through an insecure channel are monitored by the adversary, Eric then how can Ash and Ben share the key without making it appear to Eric? On first sight, it seems an impossible task but Diffie and Hellman solves this problem by developing this brilliant algorithm. [6]The Diffie-Hellman algorithm is based on the difficulty of finding discrete logarithms. For an integer b and a primitive root a of a prime number$p$, A unique exponent i can be obtained which satisfies

$b = a^{i}$ *(mod p)* where *0 d” i d” (p-1)*

The exponent i is called discrete logarithm of$b$ for the base a, mod p.

Here is how they decide the secret key of the symmetric cipher over an insecure channel. Fig.2.2 has been taken from https://wizardforcel.gitbooks.io/practical-cryptography-for-developers-book/key-exchange/diffie-hellman-key-exchange.html. It gives a brief summary of Diffie-Hellman key exchange algorithm. [5]At first a large prime$p$ and a primitive root g of prime p is chosen by Ash and Ben mutually. They are made public. Ash selects an integer a which he keeps as a secret, similarly Ben selects an integer b which he does not share with anyone. With the help of their random integers,

Ash calculates $A = g^{a}(\text{mod}\,p)$ and
Ben calculates $B = g^{b}\,(\text{mod}\,p)$.

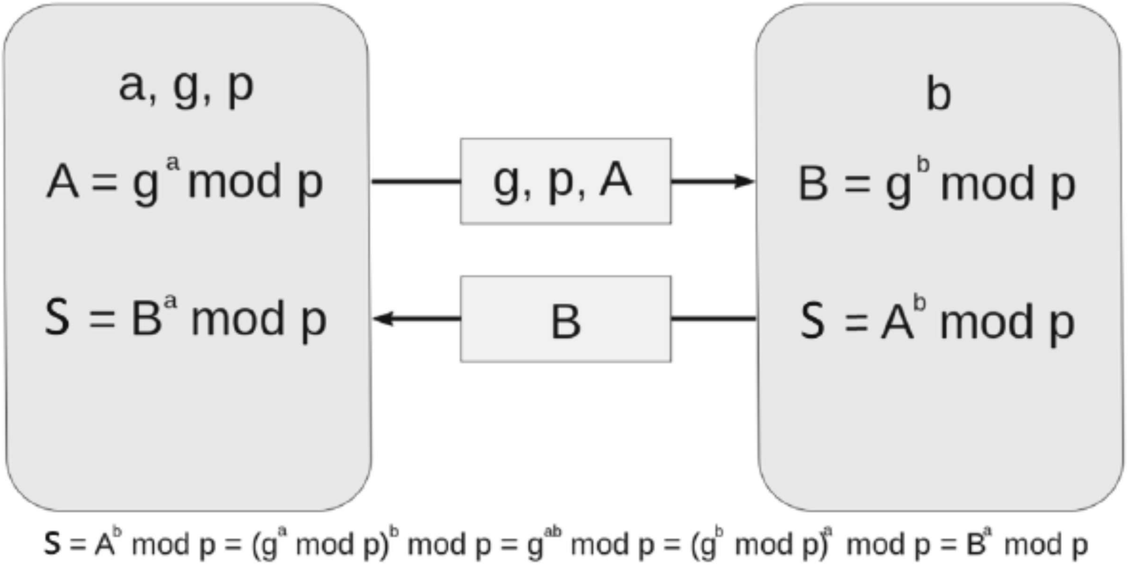

a, g, p
A = g^a mod p
S = B^a mod p
g, p, A
B
b
B = g^b mod p
S = A^b mod p
S = A^b mod p = (g^a mod p)^b mod p = g^ab mod p = (g^b mod p)^a mod p = B^a mod p

Figure 2.2 The Diffie Hellman Key exchange algorithm

Hereafter Ash sends$A$ to Ben and Ben sends$B$ to Ash. They again use their random integers to calculate the desired secret key.

Ash computes S = B^{a} (mod p) and
Ben computes S = A^{b} (mod p)

Thus, they have successfully developed the key using an insecure channel.

Suppose that adversary Eric has noted all the messages being transmitted so he hasp, g, A, B with him. To find the secret key, he will try to solve$g^{a} = A$ (mod p) for the value of a then she would have to solve $g^{ab} = (g^{a})^{b} = S$ for the value of b. [7]Solving these equations is equivalent to finding discrete logarithms which is presumed to be very difficult. If he can solve these equations, then he can successfully breach the system.

Attacks on Diffie-Hellman key exchange algorithm

- Brute Force Attack: Longer key lengths make it difficult to find the secret key by the brute force attack.
- [6]Man-in-the-Middle Attack: In this attack, the adversary Eric does the follows4.

1. Eric selects two secret integers x_1 and x_2 and then calculates the corresponding public keys y_1 and y_2 .
2. Ash sends A to Ben.
3. Eric interrupts this message and sendsy_1 to Ben. Eric computes
   $K^{'} = (A)^{x}{}_{2}$ mod p
4. Ben receives$y_1$and computes K = $(y_1)^{b}$mod$p$.
5. Ben sends B to Ash.
6. Eric interrupts this message and sends$y_2$to Ash. Eric computes
   $K = (B)^{x}{}_{1}$ mod p
7. Ash receives$y_2$and computes $K^{'} = (y_2)^{a}$ mod$p$

Now, Ash and Ben think that they have shared the secret key but instead Ash has shared the secret key $K^{'}$ with Eric and Ben has shared the key K with Eric. Hereafter Eric can read all the messages exchanged between Ash and Ben by means of the following manner.

1. Ash transmits an enciphered version of message M i.e. E($K^{'}$,M).
2. Eric checks this ciphertext, deciphers it and obtains M.

3. If Eric just want to spy the communication between Ash and Ben then he would send E(K,M) to Ash but if he wants to alter the messages being sent to Ben then he would send E(K, N) , where N is the modified message.

It is impossible to detect that such kind of attack has taken place because this key exchange algorithm does not provide authentication to the users. This attack can be prevented by using digital signatures and public key certificates. It is time consuming and involves too many computations. Diffie-Hellman algorithm can not be used to sign digital signatures.[4] It can be used to transmit the key of symmetric encryption. Almost every encryption technology uses Diffie-Hellman algorithm to increase their security, couple of them are Secure Socket Layer (SSL), Secure Shell (SSH), Internet Protocol Security (IPSec), Public Key Infrastructure (PKI), Internet Key Exchange (IKE) and all those other things that relies on these protocols.

2.3 Digital Signature Algorithm

As the world is growing more interest towards internet for its transactions and business activities,it becomes equally important to guard the content from unintended access. A digital signature is a computerized variant of traditional signature. [6]Transporting digital signature is a piece of cake. It cannot be copied easily. Modification of message content is not possible until and unless an illegitimate user has the private key of sender.

Fig. 2.3 has been taken from the website https://www.tutorialspoint.com/index.htm. It shows an approach to obtain the digital signature for a message. Digital signature for a message can be obtained using “Hash Function” and sender’s private key. Input to the hash function is the message which is being signed. Hash function compresses the message to a fixed size. Compressed message is given as an input to the signing function and the output of the signing function is the digital signature of the desired message. Sender append this signature with the corresponding message he is going to send so that receiver can make sure that the message has been sent and signed by the intended user. At receiver’s side, message is again passed as an input to the hash function and the hash value of the received message is calculated. Receiver verifies the signature using a verification algorithm and the public key of sender which produces a hash value. If the computed hash value matches with this hash value then the signature is assumed to be verified. [6] The difficulty to compute discrete logarithm is the basis of DSA.

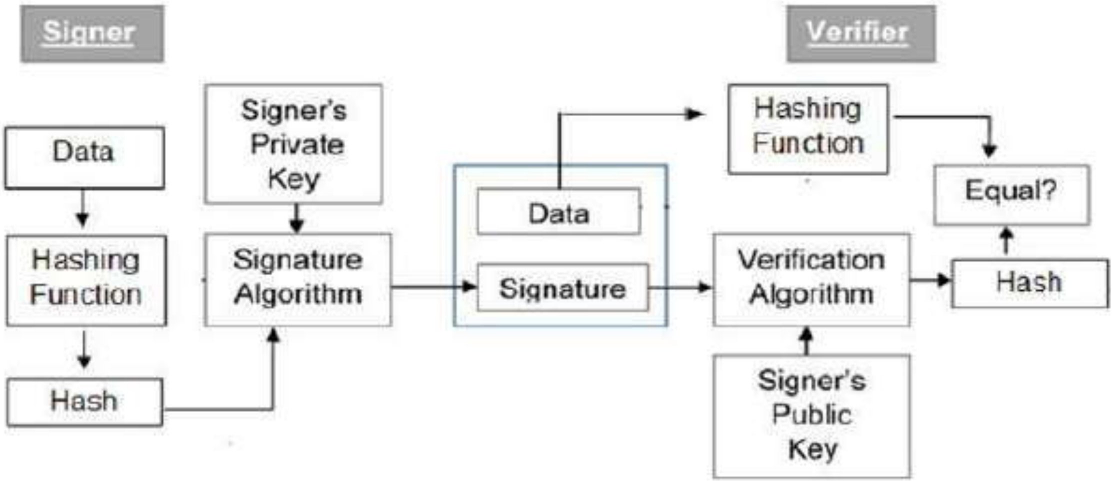


Figure 2.3 Approach to a Digital Signature

Choose a prime number q of length 160-bit. A prime number p of length in between 512 - 1024 bits is chosen in such a manner that $q|(p-1)$. Select $g = h^{(p-1)/q}$ (mod p) where 1 d” h d” (p-1). Note that $g > 1$. These are the global parameters for the members of the group.

Now, each member of the group chooses their private key and develops the public key. A random integer x, where $0 < x < q$, is chosen as the private key. $y = g^x$ (mod p) is the public key. Given the value of x, it is easy to find y but doing the converse is equivalent to solving discrete logarithm which is presumed to be difficult.

To sign a message *M*, sender follows the following procedure:

1. A randomly or pseudorandomly produced integer *k* is chosen by the sender. For every time the sender signs a message, he should choose a different value of *k.*
2. Calculate $r = (g^{k} \text{ mod } p) \text{ (mod } q)$.
3. Calculate $s = [k^{-1} \text{ (H(M) } + xr) \text{] mod } q$ Where H(M) is the hash value of message M.
4. (r , s) is the sign of sender for the message M. The sender appends his sign with the message M and sends this concatenated text.

To verify the message M, receiver follows the following procedure:

1. Receiver obtains the value of (p, q, y, g) from the public domain.
2. Calculate $u_{1} = [\text{H(M) } w] \text{ mod q}$ where $w = s^{-1}$ mod q.
3. Calculate $u_{2} = rw \text{ mod } q$.
4. Calculate $v = [(g^{u}_{1\,.}\, y^{u}_{2}) \text{ mod } p] \text{ mod } q$.
5. The sign is proved to be true if the value of *v* and *r* is same.

Let us see how this verification proves that the message has been sent by the intended sender. If we focus at how s was calculated, we will observe that

$\text{H(M)} = (ks - xr) \text{ mod } q$

Which hints that,

$$s^{-1} \text{ H(M)} = (k - s^{-1}xr) \text{ mod } q.$$

Thus, $k = (s^{-1} \text{ H(M)} + s^{-1}\, xr) \text{ mod } q$

i.e. $k = (u_{1} + xu_{2}) \text{ mod } q$.

Therefore, $g^{k} = g^{u}_{1}{}^{+\,a\,u}_{2} = [(g^{u}_{1\,.}\, y^{u}_{2}) \text{ mod } p]$ mod and we have $v = r.$

For an adversary, it is computationally impossible *q* to obtain the value of *k* from *r* and the value of *x* from *s.*

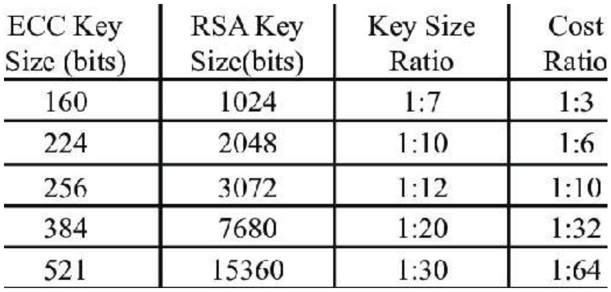
Also note that while creating the signatures, the only exponential computation required is g^{k} mod *p* and since this value is independent of the message being sent, Sender can precalculate it. [9]It is patent free therefore anyone can use it without charges.

[3]The disadvantage of DSA is that it takes a large amount of time to verify the message because the verification part is a bit computational. We cannot use this algorithm to encrypt the message being sent. Generally, SHA-1 hash function is used in the signing process so any loophole in SHA-1 algorithm will decrease the security of the digital signature.

[6]In fact, he can also compute some values of*r* in advance which he may use while signing any message. The only heavy computational task remained is the evaluation ofk^{-1} which he can again precalculate for some prechosen values of *k*. In comparison to the other digital signature, DSA needs less space. It can be used to authenticate the sender. Many encryption technologies like SSL, SSH, TLS , web servers and search engine use DSA for authentication.

2.4 Elliptic Curve Cryptography (ECC)

[6]The majority products and technologies that encrypts using public key cryptography use RSA for digitally signing the document. However, with the growing need of level of security, the key size of RSA is increasing tremendously. It puts a huge burden on the processor of those machines and applications that use RSA. This load on the processors brings unwanted consequences like that on e-business websites. [4]ECC is comparatively a new member in the class of public key encryption algorithms. [6] ECC overcomes this shortcoming of RSA. It would not be wrong if we say that ECC could be the future of public key cryptography. With a key of smaller length, it provides a great level of security as offered by RSA or any other algorithm and thus it requires less processing. [8] It has been found that the security level, offered by any other cryptosystem with a key size of 4096-bit, can be found using 313-bit key in elliptic curve cryptography. [1] Table 2.1 shows the key size ratio and cost ratio of ECC and RSA. It is quicker than any other cryptosystem. [4]Elliptic curve is defined on a particular algebraic structure where some certain operations can be carried out. A one-way function called Elliptic Curve discrete logarithm problem (ECDLP) is given by these operations. The one-way functions, which are the basis of DSA and Diffie-Hellman, is identical to ECDLP. ECDLP is used by ECC to build a cost-effective cryptosystem. [6] Unlike RSA and Diffie-Hellman, ECC is a bit complex and hard to illustrate so we confine ourselves to a rapid review of ECC.



ECC Key Size (bits)	RSA Key Size(bits)	Key Size Ratio	Cost Ratio
160	1024	1:7	1:3
224	2048	1:10	1:6
256	3072	1:12	1:10
384	7680	1:20	1:32
521	15360	1:30	1:64

Table 2.1

Key Size Ratio and Cost ratio for ECC and RSA

An elliptic curve E over the finite field, $\boldsymbol{Z_p}$ is defined as

$E : y^2 \bmod p = (x^3 + ax^2 + bx + c) \bmod p$ Where a, b, c, x and y comes from Z_p .

The set having all the pairs (x,y) satisfying above equation, is denoted by $E_p(a,b)$. It can be proved that $E_p(a,b)$, together with a point at infinity O, forms an abelian group provided that

$(4a^3 + 27b^2) \bmod p$ ‘“ 0 mod p.

An elliptic curve E over the finite field, GF(2^m) is defined as

$E : y^2 + xy = x^3 + ax^2 + b$ Where a, b, x and y belong to GF(2^m) and the computations are carried out in GF(2^m).

The set having all the pairs (x,y) satisfying above equation, is denoted by $E_2^m(a,b)$. It can be proved that $E_2^m(a,b)$, together with a point at infinity O, forms an abelian group if b ‘“ 0.

Just like we have the problem of computing discrete logarithms and finding factorization of product of two large primes, here we have the difficulty to obtain the value of k for the given value of kP Where k is a positive integer and it is less than p.

2.4.1 ECC Diffie-Hellman Key Exchange Algorithm

[6]Ash and Ben need to share a secret key. They can successfully share the secret key by using the procedure given below :

1. They begin with mutually agreeing on a number q which is either a large prime p or has the form 2^m then they choose the value of a and b.
2. Second, they choose the base point $G = (x_1, y_1)$ from the elliptic curve having a large order say n.
3. Ash chooses a number n_A such that $n_A < n$. It is the private key of Ash. He computes his public key using the formula $P_A = n_A \cdot G$.
4. In a similar way, Ben chooses his private key n_B and then calculates his public key $P_{B.}$
5. Both sends their public keys to each other.
6. Ash calculates the secret key $k = n_A \cdot P_B$ and Ben calculates the secret key $k = n_B \cdot P_{A.}$

It is easy to observe that they both have shared the same key. An intruder has to find the value of k for the given values of G and kG which is a very tough task.

2.4.2 Elliptic Curve Encryption/Decryption

[6]We can use the theory of elliptic curve to develop many encryption/decryption techniques. Here we will discuss the simplest one. The plaintext P_m is represented as an ordered pair. Users begin with mutually agreeing on a number q which is either a large prime p or has the form 2^m then they choose the value of a and b. Second, they choose the base point G from the elliptic curve. These are called global parameters. They are declared in public domain and they will be used while performing encryption/ decryption. Every user A chooses his private key n_A and computes the public key $P_A = n_A \cdot$ G.

An integer k is chosen by the sender secretly, Ash. He then encrypts P_m and obtains the enciphered text C_m Where

$C_m = [k\text{G}, P_m + k\,P_B\,]$

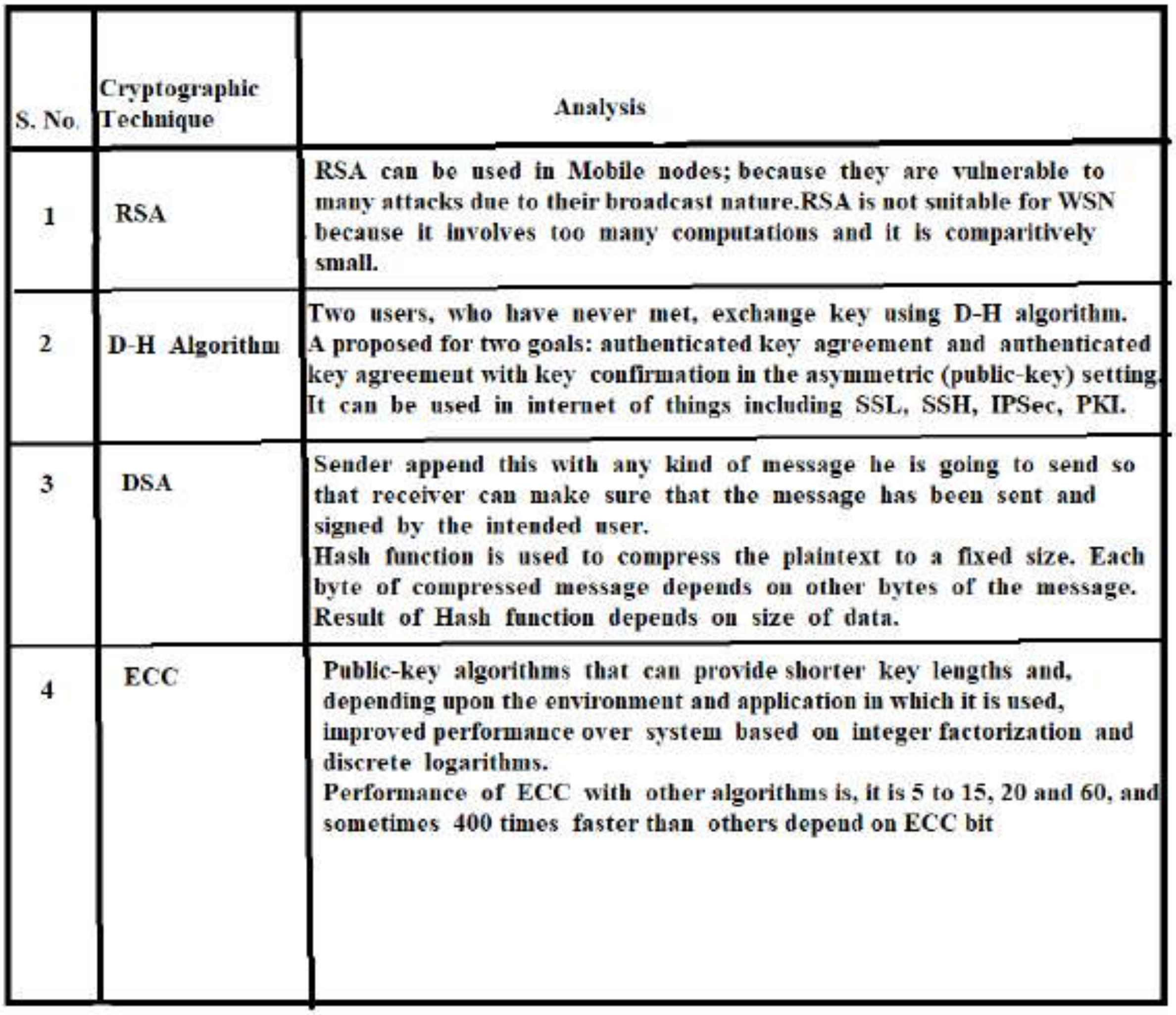
Observe that the public key P_B of receiver, Ben has been used while performing the encryption. To decipher C_m , Ben multiply the first component of C_m with his private key n_B and then subtract the resultant from the second component of the ciphertext. Thus, he obtains the plaintext P_m.

Mathematically,

$P_m + k\,P_B - (n_B \cdot k\text{G}) = P_m + k\,P_B - k\,P_B = P_m$.

[4][7]ECC is used in master cards, mobile phones, internet of things, bitcoins businesses, sensors etc. The security offered by ECC depends on the difficulty faced by the intruder in finding the value of k for the given values of kP and P. ECC encryption/ decryption technology is reasonably secure. No such concrete attacks have been found till now. There are few which can be reduced using different technologies. However, there is nothing like perfection therefore ECC should be carefully enacted.

2.5 [4]*Analysation*



S. No.	Cryptographic Technique	Analysis
1	RSA	RSA can be used in Mobile nodes; because they are vulnerable to many attacks due to their broadcast nature.RSA is not suitable for WSN because it involves too many computations and it is comparitively small.
2	D-H Algorithm	Two users, who have never met, exchange key using D-H algorithm. A proposed for two goals: authenticated key agreement and authenticated key agreement with key confirmation in the asymmetric (public-key) setting. It can be used in internet of things including SSL, SSH, IPSec, PKI.
3	DSA	Sender append this with any kind of message he is going to send so that receiver can make sure that the message has been sent and signed by the intended user. Hash function is used to compress the plaintext to a fixed size. Each byte of compressed message depends on other bytes of the message. Result of Hash function depends on size of data.
4	ECC	Public-key algorithms that can provide shorter key lengths and, depending upon the environment and application in which it is used, improved performance over system based on integer factorization and discrete logarithms. Performance of ECC with other algorithms is, it is 5 to 15, 20 and 60, and sometimes 400 times faster than others depend on ECC bit

2.6 Conclusion

After making thorough analysation of the aforementioned cryptographic techniques, we came to the following conclusions:

· RSA is one of the most popular and strongest algorithms in the family of public key algorithm. It can be used to perform encryption/decryption, to create digital signatures and to share a secret key between two parties. Till now no such concrete attack have been found which can break this algorithm but because of the longer key size, It is a bit slow therefore it might bring burden on the processors of the machines in which they are used.

· Diffie-Hellman algorithm is one of the simplest algorithms in the family of public key algorithm. Almost every encryption technology uses Diffie-Hellman algorithm to increase their security, couple of them are Secure Socket Layer (SSL), Secure Shell (SSH), Internet Protocol Security (IPSec), Public Key Infrastructure (PKI), Internet Key Exchange (IKE) and all those other things that relies on these protocols. It can be used to transmit the keys of symmetric encryption, but It cannot be used to create digital signatures. It cannot be used to perform encryption/decryption as well. It involves too many computations therefore it is time consuming and Since it does not provide authentication, it is prone to Man-in-the-middle-attack. This attack can be overcome by using digital signatures which will authenticate the two parties in front of each other.

· [9]DSA produces the signature of shorter length in contrast to other digital signature standard. When the two communicating parties are not trusted by each other, In that situation DSA can be used. It is quick. It takes lesser storage. It is patent free therefore anyone can use it without charges. Many encryption technologies like SSL, SSH, TLS , web servers and search engine use DSA for authentication. The disadvantage of DSA is that it takes a large amount of time to verify the message because the verification part is a bit computational. We cannot use this algorithm to encrypt the message being sent. It cannot be used as a key exchange algorithm as well.

· With a key of smaller length, ECC provides a great level of security as offered by RSA or any other algorithm. ECC takes less storage. RSA is slower than ECC as It involves too many calculations. ECC is used in master cards, mobile phones, internet of things, bitcoins businesses, sensors etc. No such concrete attacks have been found till now on ECC but it is a bit complex than RSA and Diffie-Hellman algorithm.

References

1. Bai T, Daisy & Rabara, S. & Jerald, A. (2015). *Elliptic Curve Cryptography based Security Framework for Internet of Things and Cloud Computing.* International Journal of Computer Science and Technology [IJCST]. 6. 223-229.
2. Chandra M. Kota et al., "*Implementation of the RSA algorithm and its cryptanalysis*," In proceedings of the 2002 ASEE Gulf-Southwest Annual Conference, March 20 – 22, 2002
3. Educba *https://www.educba.com/digital-signature-algorithm/*
4. Jirwan Nitin, Singh Ajay, Dr. Vijay Sandip. (2013). *Review and Analysis of Cryptography Techniques.* International Journal of Scientific & Engineering Research Volume 4, Issue-3 March 2013.
5. Mishra, Manoj & Kar, Jayaprakash. (2017).*A study on Diffie-Hellman key exchange protocols*. International Journal of Pure and Applied Mathematics. 114. 10.12732/ijpam.v114i2.2.
6. Stallings Williams.(1999).*Cryptography and network security: Principles and practice.* Upper Saddle River, N.J: Prentice Hall.
7. Stolbikova Veronica (2016) *Can Elliptic Curve Cryptography be Trusted? A Brief Analysis of the Security of a Popular Cryptosystem*. ISACA JOURNAL VOL 3
8. Trappe, Wade and Washington, Lawrence C. *Introduction to Cryptography with Coding Theory* (2nd Edition) 2005 Prentice-Hall, Inc. USA
9. Website *https://www.educba.com/*