Volume 5, Issue 4, 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Comparative study for Symmetric & Asymmetric Cryptography for Passive RFID Tags

Julietta John
M. Tech. Scholar,
Department of Computer Science & Engineering,
School of Engineering & Technology,
Sharda University, Gr. Noida (U.P.), India

Gouri Sankar Mishra
Assistant Professor,
Department of Computer Science & Engineering,
School of Engineering & Technology,
Sharda University, Gr. Noida (U.P.), India

Abstract— Radio Frequency Identification (RFID) is a technology to identify objects or track assets and has received many applications over the years. Many lightweight cryptographic algorithms have been suggested over the time. Till recently, passive radio-frequency identification was considered weak to implement public-key cryptography. To push the application range asymmetric cryptographic primitives have to be explored. In this paper a system architecture which enables a two level security is proposed. Both public key cryptography and private key cryptography using Elliptic curve cryptography (ECC) and Advanced Encryption Standard (AES) is used to derive a comparative study. The transmission of data requires a secure authentication protocol using Advanced Encryption Standard.

Keywords— RFID, Security Threats, AES, ECDLP, Authentication.

I. INTRODUCTION

RFID System known as Radio Frequency Identification is a technology introduced for automatic identification of objects or people. There are mainly three components that complete the whole system which are a tag which can be active, passive or semi-active second is a reader and the last component is the back-end database or also known as the middleware. Figure 1 gives a generalized idea about the working of the RFID network. The main advantage of the RFID system is that it is very cost effective. Its primary component the tag is a tremendously low-cost and tiny device. An RFID reader is technically called as a transceiver or interrogator. These RFID readers are devices that read or write data to or from RFID tags. The RFID middleware plays a very important role in converting the low-level RFID hardware information into useable event information. The tags comprise of an antenna and an integrated circuit which perform the function of transmitting data to the RFID reader. The radio waves are then converted into a more useable form of data by the reader. All the information that is collected from these tags by the interrogating readers is transmitted through a communication channel to the host computer system. The data that is transferred can be stored in a database and analysed whenever required henceforth.

Classification of the RFID tags that are widely used is done in three ways: Operating Frequency, Powering Techniques and Memory type. Electromagnetic Frequency is the primary source by which all the operations are conducted in the RFID System. The electromagnetic spectrum which powers every operation in the RFID system is basically divided into four parts. 124 kHz to 135 kHz is the operational field of the Low Frequency tags almost up to a range of half a meter. 13.56 MHz is the operational field of the High Frequency Tags almost up to a range of a meter or more. 860 MHZ to 960 MHz is the operational field of the ultra high frequency tags which range up to 10 meters [4]. Depending on the powering technique that is used RFID tags can be broadly classified into three classes. Passive Tags are only powered by the signal from the interrogating reader. Therefore, we can say that all the power that is required is provided by the reader. Active tags have on board power unlike passive tags. Active tags are functional by their own on board power source for the operations of the tag over a period of time. A semi active tag can be defined as a combination of a passive tag and an active tag. Whenever the semi-active tag enters the electromagnetic field of the interrogating reader the passive component of the tag is energized. When the tag is energized the active component of the tag is comes into action which sends an RFID signal. So the battery of the tag plays it role when it is activated by the passive parts of the tag. After a programmed quantity of time the battery goes into sleep mode. Another group of RFID tags can be constructed based on its memory type. The memory space in the tag can be used in both forms, it can be used as writeable and non – writable data storage [4]. The tags are contrived in this manner, they are either read only or write once read many or fully rewritable. In read only tags as the name suggests the communication between the reader and the tag is unidirectional and the tags can only be read by the reader. The read write tags provide the facility of both reading the tag information and also writing information to the tag at any time. To store the information and to send it to the reader the tag has a memory space.

II. THE UNDERLYING ATTACKS OF THE RFID SYSTEM

We can divide the RFID system into four layers: Physical Layer, Network Transport Layer, Application Layer, and Strategic Layer. The Physical Layer constitute of the hardware mostly such as the RFID Tags, the Radio Frequencies and the RFID Reader. The Network-Transport Layer compiles the protocols that are being used. It includes examples such as ISO standards and other protocols. The Application Layer comprises of the middleware or the backend database that is being used to store all the collected information. The Strategic Layer deals with the real world constraints, the logical factors that are in role and cost versus utility exchange. Every specific layer has areas which are vulnerable to attacks [5].

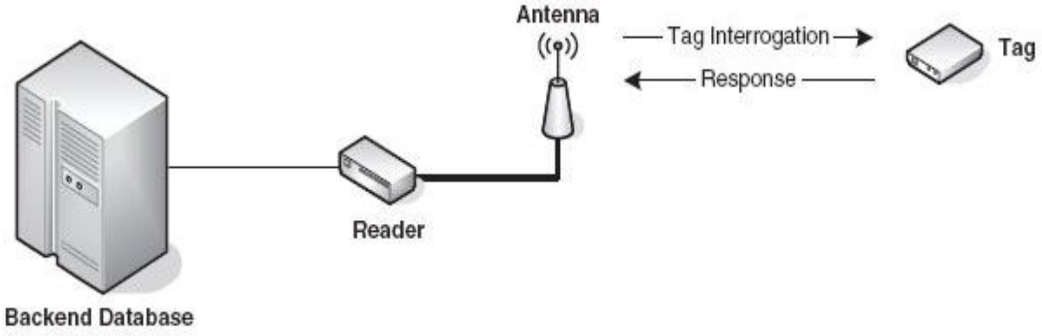

Fig. 1 Working of the RFID System

A. Physical Layer Attacks

The attacks of the physical layer deal with either temporarily or permanently disabling the tags.

- Active Jamming

In this kind of attack the adversary exploits the fact that all radio signals in the range of a tag is heard by it indiscriminately. Therefore an attacker might create a jamming by transmitting a signal in the identical range of that of the reader. In this manner the tags won't be able to continue communicating with the interrogating readers.

- KILL Command

KILL is a command which is created by the EPC Global to permanently destroy a RFID tag. According to the KILL command, using it requires an unique password which is created by the manufacturer while the tag is born. The use of this password can lead to killing the tag permanently. Even though this feature was introduced for privacy issues this feature has all the qualities to be exploited by attackers to incapacitate the communication between a tag and an interrogating reader.

B. Network-Transport Layer Attacks

The attacks that come under this layer are mostly based on the manner of the communication and on the way the information is transmitted between an interrogating tag and reader. The attacks of this layer are further classified into the attacks which target the reader, tag and the network protocols. The attacks which target the reader are eavesdropping and impersonation.

- Eavesdropping

Eavesdropping is one of the most severe and extensively deployed attacks which exploit the fact that RFID is based on a wireless nature. In this the communication between the tag and the reader is recorded with the help of an antenna by the attacker [5].

- Impersonation

In majority of the cases the communication between the tag and the reader is unauthenticated. So it becomes easy for the attacker to impersonate the identity of a reader in order to extract crucial information or to manipulate the data on stored in the RFID tags.

- Spoofing

Spoofing is more like cloning only the only difference is the presence of a physical tag. In cloning the tag is replicated but in spoofing an attacker impersonates a tag so that it can gain its privileges.

C. Application Layer Attacks

The attack which target the binding present between the tag and the user and information regarding the applications come under this classification.

- Buffer Overflow

Buffer overflows is one of the hardest security concern that persists in software and is considered a major threat. The code or data that is stored beyond the limits of the buffer which is fixed is exploited in buffer overflows. The backend database or middleware is affected when an attacker uses the RFID tags to launch these attacks. As we know that RFID tags have limited storage this attack might be considered insignificant. But there are commands that permit a tag to transmit the same data block again and again which can cause therefore buffer overflow in the middleware [5].

- Malicious Code Injection

The components of the RFID system such as the connecting network and the readers can be infected by injecting malicious code which is propagated using the RFID tags. An attacker stores the infecting code in the memory location of the tags and is further propagated using the tags itself. The feasibility of these attacks are limited which prevents this attack from being widespread. XML, JavaScript, PHP are some of the scripting languages that are being used in the middleware applications. In order to cripple the middleware an attacker can inject malicious code and affect the network[5].

III. Related Work

The area of providing solutions relevant to the threats that hover around the RFID system is a booming one. New approaches are continuously applied and finding the best that has always been a concern. Security and Privacy threats are a major concern area in the RFID network.

A. Hash Lock

The tags which use this scheme have a certain portion of their memory kept apart to store the temporary metaID. The tags are initially locked by the owner by computing the hash of the key which is selected at random. This hash output is labelled as the metaID is the one which is stored on the memory of the tag which exists in the locked state. The back end database further stores the metaID and the key. The owner queries the metaID of the tag to unlock the tag. The value which is retrieved is used to look for the key which is stored in the back end database. The key value is transmitted to the tag which performs a hash on the key and checks it with the already stored metaID. If the solutions are a match, then the tag goes into the unlock state from locked state and provides complete functionality to the readers nearby[3].

B. Randomized Hash Lock

An ID number which is transmitted and stored with the reader is allocated to the tag. Whenever queried by the reader a random number would be selected by the tag. The ID number of the tag and the random number selected by the tag would be hashed and the result would be sent to the reader. The reader would also perform a hash of the received information from the tag. On performing a hash the reader would get the ID number of the tag. By this ID number the reader can recognize the tag. A communication can be set up after the tag is recognized by the reader[3].

C. HB Protocol

This method was presented by A.Jules and S.Weis known as Hopper and Blum or HB protocol. HB protocol is classified as a challenge response protocol. Suppose Alice and a computing device C share an k-bit secret x, and Alice would like to authenticate herself to C. C selects a random challenge a 2 {0, 1}k and transmits it to Alice. The binary inner-product a • x will be computed by Alice, then the result is sent back to C. C computes a • x, and accepts it if and only if it matches its own calculation. In a single round, someone imitating Alice who does not know the secret x will guess the correct value a • x half the time. By repeating this challenge and response for r rounds, Alice can lower the probability of naively guessing the correct parity bits for all r rounds to 2−r. Alice can also inject noise into her response. The noise bit ν can be easily generated. Alice intentionally sends the wrong response with constant probability η. C then authenticates Alice's identity if fewer than ηr of her responses are incorrect[6].

D. WIPR Cryptographic Scheme

The tag is provided with the public key n and a signed unique identifier ID. The reader is provided with the private key (p, q). The reader generates a random bit string Rr ,where |Rr| = α. The tag generates two random bit strings Rt1 and Rt2, where |Rt1| = |n|−α−|I D| and |Rt2| = |n|+β. and α, β are security parameters. The reader sends Rr to the tag. The tag generates a plaintext as follows: P = Rr #Rt,1#I D, where # denotes concatenation, and then transmits the following message: M = P2 + Rt2 · n. For verification, the reader uses the private key to decrypt M. There are four candidate decryptions, so the reader checks which of the four possible decryptions contain the value of the challenge Rr it sent to the tag. If such a plaintext is found, the reader outputs the value of ID. In all other cases, the authentication fails[1].

IV. Proposed System Architecture

The UniqueID from the tag is read by the reader. After being read the data is encrypted by any of the two algorithms AES or ECC. To provide ambiguity the chosen algorithm will not be mentioned explicitly. When the data is encrypted using any of the two algorithms simultaneously time for encryption is calculated. The data is stored in the database, if the data has to be transmitted from one communicating party requires secure authentication protocol using AES.

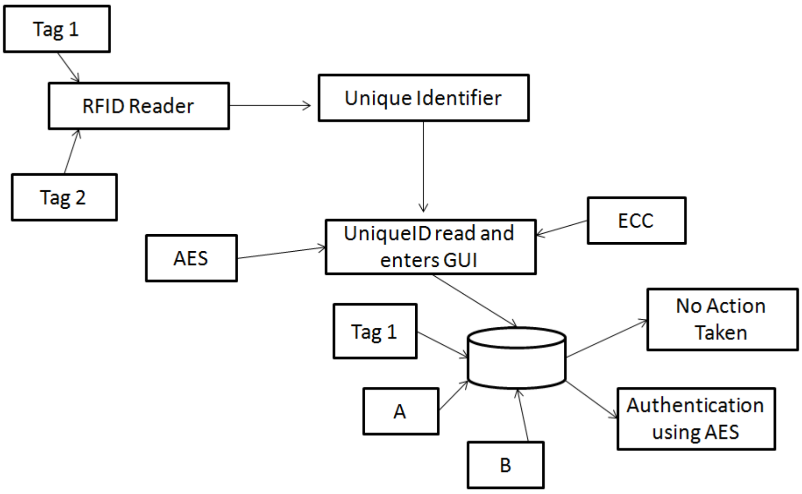

Fig. 2 Proposed System Architecture

A. Advanced Encryption Standard (AES)

AES is the standardized version of Rjindael and was added as a FIPS standard on 26 November 2001. AES is a block cipher which encrypts block size of 128,192 and 256 bits with 10, 12 & 14 rounds respectively. At the input and output we use the term data block but when this particular data block is transferred from one stage to another it is termed as state. For AES Galois Field arithmetic of 2^8 is done also termed as GF(2^8)[7]. The four main stages which are the principle elements of the AES cipher are as follows:

1) SubByte Transformation

This operation creates a substitution at every level. The main component of this stage is a look up table which is known as an s-box. The look up table is 16x16 matrix and it comprises of all the values that can possibly turn up in an 8 bit sequence. In this stage every byte present is transformed into a different byte to elevate confusion. A particular row of the S-box is traced by the left nibble of the byte and the column of the S-box is traced by the right nibble of the byte. In the decryption algorithm an inverse S-box is present which is used for the Inverse substitute byte transformation.

2) ShiftRow Transformation

This stage performs simple permutation. The first row is kept untouched while the second, third and fourth rows are shifted 1, 2 and 3 bytes respectively to the left in a circular manner. These circular shifts are performed in opposite direction in the decryption algorithm which is termed as InvShiftRows. This operation is only performed on the last three rows because in the start that is in the ShiftRows operation we did keep the first row unaltered.

3) MixColumn Transformation

Mix column transformation is the most complicated stage of AES. All the other stages provide only intra byte transformation but mix column does the actual transformation. The main role of Mix Column is to provide diffusion at the very bit level of any particular byte. The state which is the input is multiplied with a constant matrix. For S [0, 0] = [8, 7], S [1, 0] = [6, E], S [2, 0] = [4, 6], S [3, 0] = [A, 6] is written as (02 • 87) ⊕ (03 • 6E) ⊕ 46 ⊕ A6. In Galois Field the numbers are converted into a polynomial. So [0, 2] = x and [8, 7] = “$x^7 + x^2 + x + 1$”. The multiplication of these two numbers will be [0, 2] • [8, 7] = $x \cdot (x^7 + x^2 + x + 1) = x^8 + x^3 + x^2 + x$. The answer of this multiplication cannot be accepted by AES as the degree of the polynomial’s degree greater than 7. So modulo reduction is done using $M(x) = x^8 + x^4 + x^3 + x + 1$. The reduction is done as follows: $(x^8 + x^3 + x^2 + x) \bmod (x^8 + x^4 + x^3 + x + 1) = x^4 + x^2 + 1$. In this manner the other term are also multiplied and then modulo addition is performed to all the four terms.

4) AddRoundKey Transformation

This operation performs matrix addition thus increasing ambiguity and confusion after every round. In this stage there is 128 bits round key which is to be XORed bitwise with the input state which is also 128 bits. This stage deals with only column wise operation.

B. Elliptic Curve Cryptography (ECC)

The reason behind using an asymmetric cryptography is to overcome the disadvantage of symmetric cryptography. While using symmetric cryptography the key used is shared, which acts as vulnerability. But asymmetric cryptography uses a pair of public and private key. The public cryptosystems have a lot of algorithms to start with but the need for a new algorithm was present when the block size of data to be encrypted increased. To solve this concern Elliptic curve were introduced.

TABLE I COMPARING KEY LENGTH FOE DIFFERENT ALGORITHM FAMILIES

Algorithm Family	Cryptosystems	Security Level			
		80	128	192	256
Integer Factorization	RSA	1024 bits	3072 bits	7680 bits	15360 bits
Discrete Logarithm	DH,DSA, ElGamal	1024 bits	3072 bits	7680 bits	15360 bits
Elliptic Curve	ECDH, ECDSA	160 bits	256 bits	384 bits	512 bits
Symmetric Key	AES,DES	80 bits	128 bits	192 bits	256 bits

Algorithms like RSA, Diffie-Hellman, ElGamal and Digital Signature Algorithm required a key length of 1024 bits, 3072 bits, 7680 bits and 15360 bits for a block size of 80 bits, 128 bits, 192 bits and 256 bits respectively. So the motivation was to introduce an algorithm family which worked with shorter operands that is used shorter key lengths. The idea was to find another cyclic group in which the Discrete Logarithm problem was difficult compared to the modulo function. For this Elliptic curve algorithm family was introduced as seen in Table 8.1 it uses a key length of 160 bits, 256 bits, 384 bits, 512 bits for a block size of 80 bits, 128 bits, 192 bits and 256 bits respectively.

1) Elliptic Curve

The Elliptic curve over Zp where p>3 is the set of all pairs (x,y) which belongs to Zp and satisfies:

$$y^2 = x^3 + ax + b \qquad mod\ p \qquad \ldots(1)$$

Together with an imaginary point at infinity, O, where a, b belong to Zp and satisfies:

$$4a^3 + 27b^2 \neq 0 \qquad mod\ p \qquad \ldots(2)$$

Let us look at an example, as the operations performed are mod p it would be nearly impossible to plot a graph according to that. So to decrease ambiguity we consider an example where a, b belong to set of real numbers.

$y^2 = x^3 - 3x + 3$

The graph obtained for the above equation can be seen in figure 3. In the graph the symmetry is with respect to x axis. That is whatever happens on the y axis gets repeated on the y axis. In general,

$$y^2 = x^3 + ax + b$$
$$y = \pm\sqrt{x^3 + ax + b}$$

Therefore, y will get a positive and a negative value. This is the reason for the above mentioned symmetry. For a discrete logarithm problem we need a cyclic group. For a group we need:

i. A Set of elements which are points on the curve. Other cryptosystems such as Diffie Hellman uses group elements as integers. But Elliptic curve uses only points which are on the curve, which makes the approach abstract.

ii. A group operation that fulfils the group laws.

2) *Group Operations on Elliptic Curve*

The Group operations that we would understand in this section are point addition and point doubling. To get a simple and better understanding we perform the operations on real numbers rather than the set of Zp. First let us understand how point addition and point doubling is done graphically, then we can derive the analytical expression for the same. When we say point addition, P + Q = ? , we have to find the mentioned, where P and Q are both point on the elliptic curve. As we can see in Figure 3 P and Q are both points on the curve. We firstly join P and Q and extend it to the third point of intersection, which in the figure is R. This mirrored point of R on the curve is by definition P+Q. So, hence we can conclude that, P+Q= (-R), where R is the mirrored point of actual R on the curve.

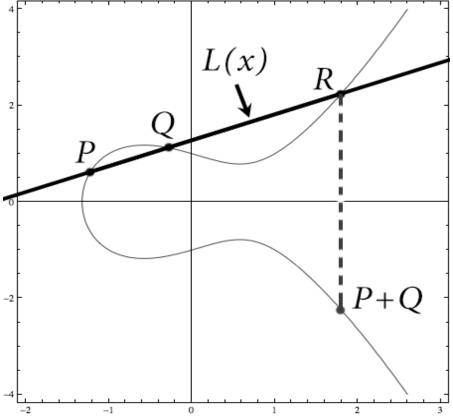

Figure 3. Point Addition

Point doubling is somewhat similar to point addition but in place of Q here we are adding P to itself. In general we can say that point doubling is P+P. So to find point doubling, the point taken in Figure 4 is P. Now we joined P and Q in the above condition. But we cannot join a point to itself here. So we construct a tangent which passes through the point P. Extend the tangent to find the point of intersection on the curve and then mirroring the point on the curve is the result of point doubling. As we notice, this is quite similar to point addition. In Figure 4 we can see the above mentioned method clearly. The point P has to be doubled, that is we have to find P+P. We draw a tangent passing through P and extend it till we find the point of intersection of the tangent on the curve. The point of intersection here is –R. Then we mirror this point which gives us R and hence the result by definition. So we can conclude that, P+P = 2P = R, where P and R lie on the curve. The graphical representation of point addition and point doubling was seen above. Now we proceed with the analytical expression for the above mentioned operations. We take two point P and Q where,

$$P = (x_1, y_1)$$
$$Q = (x_2, y_2)$$

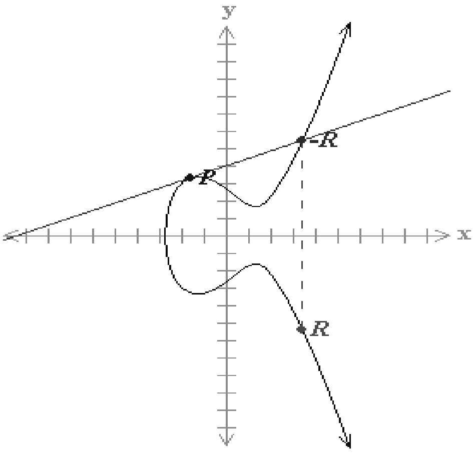

Figure 4 Point Doubling

The equation of the line, L(x) seen in Figure 3 is as follows:

$$y = sx + m \qquad ...(3)$$

Where s is the slope of the line.
To find the point of intersection we have to equate the above equations.
Equating Equation 1 and Equation 3, we get.

$$(sx + m)^2 = x^3 + ax + b$$

Solving the above equation we get,

$$s^2x^2 + m^2 + 2sxm = x^3 + ax + b$$

To find the point of intersection that is (x_3, y_3) we have to solve the above equations. As we can list that the degree of the above equation is 3, we would get 3 solutions. We already have two solutions that are P (x_1, y_1) and Q (x_2, y_2). So the point of intersection (x_3, y_3) are,

$$x_3 = x^2 - x_1 - x_2 \quad mod\, p$$

$$y_3 = s\,(x_1 - x_3) - y_1 \quad mod\, p \qquad ....(4)$$

Where,

$$s = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{Mod P, Point Addition} \\[2ex] \dfrac{3x_1 + a}{2y_1} & \text{Mod P, Point Doubling} \end{cases} \qquad ...(5)$$

3) Elliptic curve Discrete Logarithm Problem (ECDLP)

As the EC protocols depend greatly on the hardness of the Elliptic curve Discrete Logarithm Problem, it is necessary to understand what the underlying problem actually is. The point on the Elliptic curve including *O* have cyclic subgroups. Firstly, we have to understand what a cyclic group of an Elliptic curve is. By definition, the Elliptic curve Discrete Logarithm Problem states

Given is an Elliptic curve E, we consider a primitive element P and another element T. The Discrete Logarithm Problem is finding the integer d, where $1 \leq d \leq \#E$ such that

$$P + P + P + ... + P = dP = T$$

Let us look at an example to understand the Elliptic curve Discrete Logarithm Problem and the cyclic groups.

$$E: \qquad y^2 = x^3 + 2x + 2 \quad \text{mod } 17$$

For this particular equation let us take P (5, 1). We have seen in the earlier section how point doubling is done. We repeat the same operation here, while doing the point doubling we can get the cyclic group.

P = (5, 1)
2P = P + P = (6, 3)
3P = 2P + P = (10, 6) and so on, we calculate till
18P = 17P + P = (5, 16) = (5, -1) = (-P)
19P = 18P + P = (5, 16) + (5, -1) = -P + P = *O*
Further when we continue the point doubling we start getting the same results.
20P = 19P + P = *O*
21P = 20P + P = P + P = 2P

Note that these calculations are done using the equations derived in the earlier sections. We can see how a cyclic group has been generated using the point P. Basically, we are hopping on the curve itself to derive the points. Now suppose we choose a point T on the curve, the Elliptic curve Discrete Logarithm Problem is that, the number of hops done to reach T cannot be calculated. So in this case, T would act as the Public Key and d as the Private Key. Note that T will always be a point on the curve and d an integer belonging to the group Zp. In the above example we can say that the group cardinality was 19, which is denoted as n. Group cardinality or n is the number of hops that can be taken on the curve till a cyclic group is formed. The d chosen should be always less than n.

So therefore we can conclude that, all Elliptic curve protocols depend on the hardness of the Elliptic Curve Discrete Logarithm Problem.

4) Elliptic Curve Diffie Hellman (ECDH)

Elliptic curve Diffie Hellman is chosen as the asymmetric algorithm to provide two-level security for the RFID system. Elliptic curve Diffie Hellman is exactly same as the elementary Difiie Hellman, the only difference is that it uses the approach with Elliptic Curves. In other words, it is a straightforward adoption of Diffie Hellman in Zp but with the use of Elliptic curves. Firstly, let us understand the working of the protocol in two phases. The first phase is the set up phase.

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

The primitive element chosen is P which we can call as (x_p, y_p) which are points on the curve. We have two communicating parties A and B. Both A and B are aware of the two things mentioned above. Firstly, A chooses a private key for itself which satisfies the condition, $1 \leq a \leq n-1$. Similarly B also chooses a private key for itself which satisfies the condition, $1 \leq b \leq n-1$.

A:

$$a = K_{private}\, A \qquad ,where\ 1 \leq a \leq n-1$$

B:

$$b = K_{private}\, B \qquad ,where\ 1 \leq b \leq n-1$$

Now the private keys have been selected for both A and B. Note that the Private Key is always an integer. The next step is to compute the Public Keys which are transmitted. The Public key is calculated by performing scalar multiplication. For in case of A, the public key is computed by scalar multiplication of a.P, where P also called as the Generator point was the primitive element we choose in the first phase. Similarly it is done also for B and the public keys are computed.

A:

$$A = K_{public}\, A$$
$$A = (K_{private}\, A)P$$
$$A = a \cdot P = (x_{A\,,} y_A)$$

B:

$$B = K_{public}\, B$$
$$B = (K_{private}\, B)P$$
$$B = b \cdot P = (x_B, y_B\ )$$

Here both the public keys A and B are points on the curve. This public key is transmitted to either parties and a common key is derived which can be used to encrypt the data.

A:

$$a \cdot B = (x_{AB}, y_{AB})$$

B:

$$b \cdot A = (x_{AB}, y_{AB})$$

Now as we can see both A and B have arrived at the same conclusion (x_{AB}, y_{AB}) . They can choose either x_{AB} or y_{AB} to encrypt the data.

C. Unilateral Authentication

If only one party has to be authenticated then unilateral authentication is used. Authentication means that an object proves its claimed identity to its communication partner. In Unilateral authentication, there are two communicating parties A and B. Both posses the same private key K, B sends a random number to A. A encrypts the random number with the shared key K and sends it to B. B proofs the result and verifies A[2].

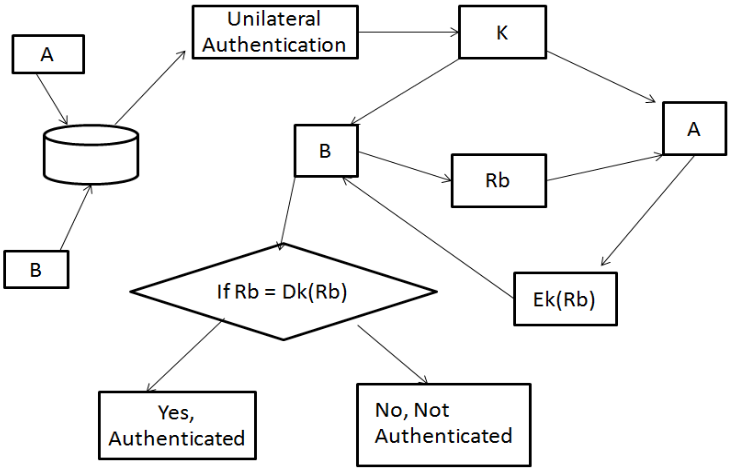

Figure 5 Unilateral Authentication

D. Mutual Authentication

In Mutual authentication, B sends random no to A, A encrypts R_B and a self generated random number R_A with shared key K and sends it to B. B decrypts the message and can proof if R_B is correct and gets R_A. B changes the sequence of the random numbers encrypts it with K and sends it to A. A proofs the result and verifies the identity of B[2].

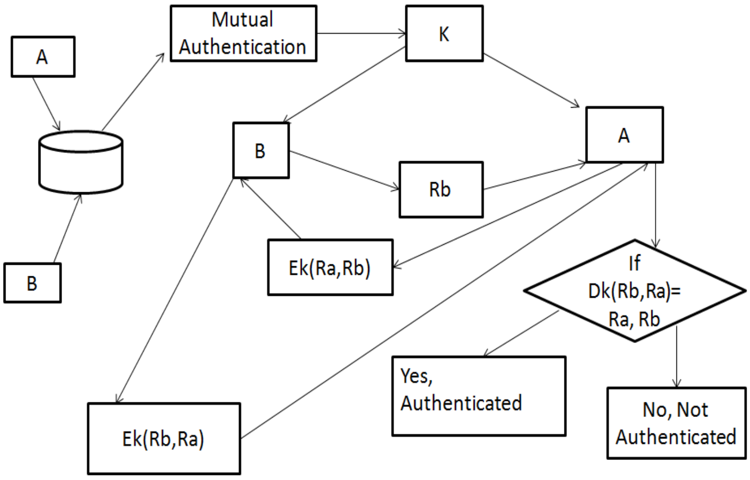

Figure 6 Mutual Authentication

V. Result and Conclusion

The execution time of both the above mentioned algorithms have been calculated and plotted. According to the graph, we can see that AES takes more time compared to ECDH which further makes ECDH a better choice as it also provides better security.

TABLE 2 TIME CALCULATED FOR TWO DIFFERENT ALGORITHM FAMILIES

Algorithm Family	Cryptosystem	Data Length	Key Length	Time(ns)
Symmetric Key	AES	128 bits	128 bits	2500
Elliptic Curve	ECDH	128 bits	256 bits	2000

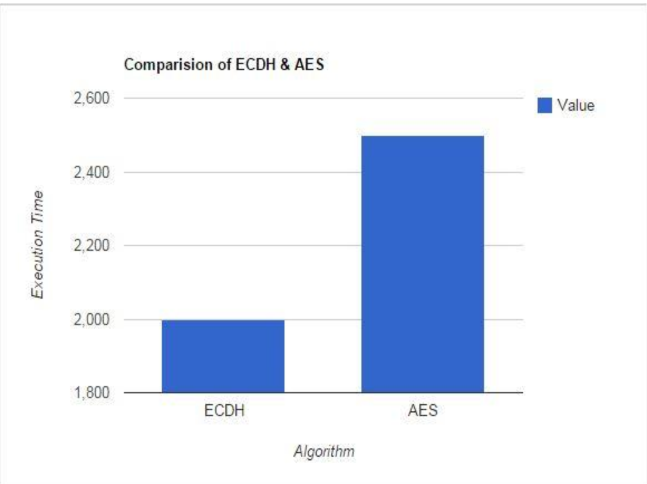

Figure 7. Graph showing Performance Analysis

References

[1] Alex Arbit, Yoel Livne, Yossef Oren, Avishai Wool , "Implementing public-key cryptography on passive RFID tags is practical", Springer, April 2014.

[2] Martin Feldhofer," A Proposal for an Authentication Protocol in a Security Layer for RFID Smart Tags", Institute for Applied Information Processing and Communications (IAIK), Austria.

[3] Stephen August, "Security and Privacy in Radio-Frequency Identification Devices" , Weis Massachusetts Institute Of Technology May 2003.

[4] Monzur Morshed, "Effective Protocols for Privacy and Security in RFID Systems Applications", PhD thesis, Staffordshire University, March 2012.

[5] Aikaterini Mitrokotsa, Melanie R. Rieback and Andrew S. Tanenbaum, "Classification of RFID Attacks", Department of Computer Science, Vrije Universiteit, Netherlands.J.

[6] Ari Juels and Stephen A. Weis , "Authenticating Pervasive Devices with Human Protocols", Advances in Cryptology -- CRYPTO 2005, Presentation Slides LNCS, volume 3621, pages 293-308, 2005

[7] Behrouz A Forouzan, "Cryptography & Network Security".