3rd International Conference and Workshops on Recent Advances and Innovations in Engineering, 22-25 November 2018
(IEEE Conference Record # 43534)

A Comparative Analysis of Cryptographic Keys and Security

Vishal Choudhary
Research Scholar
Banasthali University, Jaipur,India
vishalhim@yahoo.com

Dr S.Taruna
Associate Prof.
JK Lakshmipat University, Jaipur
staruna@yahoo.com

Lal Babu Purbey
Assistant Professor,
Poornima College of Engineering, Jaipur
lalbabu.purbey@poornima.org

***Abstract*— In wireless sensor networks (WSNs), security has a fundamental significance. With advancement in use of internet of Things (IOTs) applications there are challenges to empower the sensor nodes with secure and lightweight algorithms that can run efficiently with minimum energy consumption as well as to provide the security comparable to traditional computer networks. In this paper, the analysis is concentrated on the key generation techniques, encryption/decryption solutions. We have briefly discussed about security algorithms with their limitations in sensor network. We also presented the importance to emerging technology for providing security in wireless sensor networks.**

Keywords—Cryptography, Cryptographic keys, Key Generation Algorithms, ECC

I. INTRODUCTION

The architecture of the crypto algorithm directly related to its strength. An algorithm is considered to be efficient if it is up to the security benchmark of a system. The comparative analysis of cryptographic algorithms is shown in table 3.This paper discussed various key generation techniques and cryptographic encryption/decryption solutions. We have briefly described the different algorithms and have given more importance to emerging technology for providing security based on elliptic curve cryptosystems, there is a brief review on symmetric and asymmetric key encryption also discussed secure hash algorithms. We also discussed different techniques for random number generation which is initial input parameter for different encryption key at the consecutive interval of time. We also did a comparative analysis of different algorithms used in the security system. In section 1 we discussed symmetric encryption with a real-world example. In section 2 we discussed characteristics of cryptographic keys. in section 3,4 we discussed key derivation and key generation algorithms. Section 5 and 6 we discussed key generation, key size, and key ring. in section 7,8 we discussed the Random number and their characteristics and cryptographic hash function. in section 9 and 10 we discussed elliptical curve cryptography and comparison in between different cryptographic key generation algorithms.

II. SYMMETRIC ENCRYPTION TECHNIQUE

The common problem of security, confidentiality, the integrity of contents during transmission and distribution of information over the network [24]. These problems still exist in our daily life. Let we consider an examiner wants to send the students marks to result branch, he can personally submit the student's marks in result branch then, there is no problem of security, confidentiality, the integrity of the content. But if the examiner is another city then personally delivers the students marks is a difficult and time-consuming process. So there are different options available with the examiner.

a. The simplest solution is that examiner puts the student's marks in a confidential packet, seal it, and send by postal service. There is a probability that nobody can open before it arrived in result branch.

b. The second option is to send the students marks via a hand –delivery method.

c. The third option is examiner put the marks inside a box and seals it with a very secure lock and sends the box to result branch.

The option a. is the best solution but it doesn't give 100% guarantee that no one can open the envelope during transit, where there is a possibility, a person can open the envelope read or modify the marks seal back the envelope and delivers to result branch. This process violates the concept of security, integrity, and confidentiality of contents. Option b is still secure but it does not give a full guarantee that content is delivered without any security violation. Option c is the best method of delivery but this gives rise to another problem. How to deliver the key, if the key is sent along with box then the whole idea of security will fail; now anyone can open the box with key. Another solution is to send the key through a different channel, first, send the box and then send key, its best solution. This will give birth to another problem if examiner wants to send marks to different branches then he will need more lock and key pairs and to manage all the process is somewhat difficult. As an outcome, we found that no solution is fully adequate. Either it is not perfect or practically not feasible. Option-c gives rise to the problem of key distribution and management.

In Network security and cryptography, there are two class of encryption known as symmetric key encryption and asymmetric key encryption. in symmetric key encryption both communicating parties share the same key, but in asymmetric key encryption there are two different types of key one is known as a public key and another is known as a private key. Key transportation is a disadvantage of symmetric key encryption. From the sender to the receiver the secret key is to be transmitted before actual message transmission. And if some tap and capture the key the communication become insecure. While in public key encryption sender encrypts the message using public key and receiver decrypt the message using the private key. Only the entity which has private key corresponding to public key can decrypt the message. Suppose if in public key encryption number to an entity in communication is n, the number of key- lock pairs will be (n (n-1)/2).

978-1-5386-4525-3/18/$31.00 ©2018 IEEE

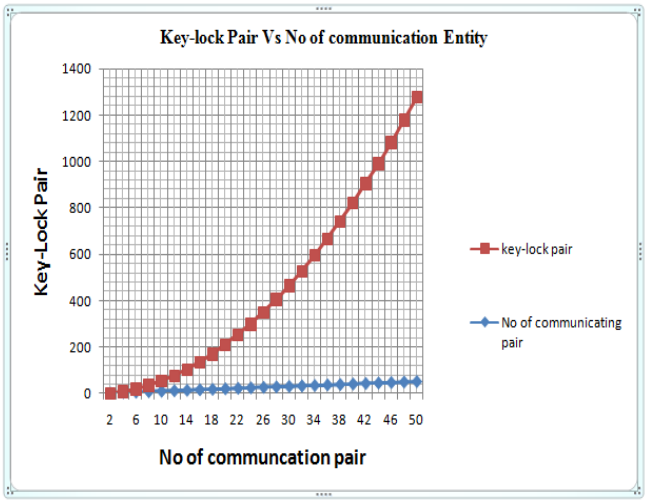

Fig 1. Key-lock pair vs no. of communication entity

As the As the number entities involved in communication increases from the graph in figure 1. We conclude that the number of key lock pair increases drastically and to manage such a huge key-lock pair give rise to the problem of key distribution and management and we need a trusted third party or system T of managing it. And each communicating pair has to converse with T to get the lock-key pair. Therefore, T must itself be trustworthy and accessible to all.

III. Characteristics of Cryptographic Keys

A value used along through a cryptographic algorithm that establishes its procedure in such a technique that an entity with information of the value can reproduce or reverse the process, while an entity without the information of the value cannot do. This value is known as a key. Cryptographic keys are used for following operations.

i. The conversion of plaintext into cipher text.

ii. The translation of cipher text into plaintext.

iii. The calculation of a digital signature from information.

iv. The authentication of a digital signature,

v. The calculation of an authentication policy from facts.

vi. The computation of a mutual secret that is used to deduce keying method.

vii. The seed of extra keying matter from a key-derivation key.

Key generation strategies took into consideration the following points:

a. It is algorithmically simple to generate a pair of public key and private key.

b. It is algorithmically simple to generate a cipher text using the public key.

c. It is algorithmically simple to decrypt the cipher text using the private key.

d. It is not possible to deduce the private key from the public key.

e. It is not possible to deduce the message from the cipher text and the public key.

IV. Key Derivation

A key derivation function [9] is a basic module of the cryptographic system. It is a process of generating various keys from a shared secret password to secure a communication session. Its aim is to take a basis for primary keying material which should be uniformly random. on the other hand, when the primary key stuff is not consistently random then KDF needs to dig up from this defective source a initial pseudorandom key from which additional keys can be derived using pseudorandom function(PRF) [27].key derivation function pursue normal extract then expand method. The key derivation function can be expressed as follows: DK=KDF(secret key, Ran, iteration). The derived key DK is expressed in the form of key derivation function DKF which is manipulated with the aid of secret key that is original shared secret, random number as a seed and the total number of iteration used for generating a derived key.

V. Key Generation Algorithms

In order to deal with key generation and key exchange, different types of algorithm were developed .some of them are discussed below.

A. Diffie- Hellman Key Exchange

It is a key agreement protocol. The beauty of this technique is that the two entities, who want to converse securely, can make agreement on a symmetric key using this method. it obtains its protection from the complexity of calculating discrete logarithms in a finite field, in contrast with the effortless calculation of exponentiation in the same field. In this protocol, if two parties want to communicate with each other will agree on two huge prime number P1and P2. And these numbers need not be kept back undisclosed. Now first party e.g. A choose another huge random number X and calculate a value for A with the formula: $A=P2^X$ mod P1. Now the first party transmits this calculated value of A to second party B. now the second party choose another large random integer Y and calculate a value $B=P2^Y$ mod P1. then transmit this calculated value of B to first party A. now the first party calculates the secret key K1 as follows.$K1=B^X$ mod P1. Second party B calculates the secret key K2 as follows: $K2=A^Y$ mod P1. we found both K1=K2=K summery of the calculation is shown below.

TABLE I. SECRET KEY CALCULATION

X	Y	P1	P2	$A=P2^X$ MOD P1	$B=P2^Y$ MOD P1	$K1=B^X$ MOD P1	$K2=A^Y$ MOD P1
3	6	11	7	2	4	9	9

In fact, both parties exchange P1, P2, A, B, based on these values X know to the first party and Y known to only the second party cannot be calculated easily. Mathematically to calculate X and Y is too complex and if they are extremely large numbers, an attacker cannot calculate X and Y, and therefore cannot drive K.

Can we come to the conclusion that the Diffie-Hellman key-exchange algorithm solves all problems associated with key exchange? The answer is NO.

Diffie-Hellman key exchange algorithm can be broken by man-in-middle-attack which is also known as the bucket-bridge attack.

A man-in-the-middle attack is a process where a malicious actor includes him/herself into a conversation between two parties, imitate both parties and gains access to information that the two parties were trying to send a message to each other.

If one party want to communicate securely with a second .then the first party want to do a Diffie-Hellman key exchange with the second. for this first party send the values of P1 and P2 to second .these values form the basis to calculate K1=K2=K. the first party does not know the man in the middle is listening quietly to the conversation in between both parties. Men in the middle simply pick up the values of P1 and P2 and also forward them to second as the original. Now, let us assume that first, second and man in middle selected random number X and Y.

TABLE II. DIFFIE-HELLMAN KEY EXCHANGE WITH MAN IN MIDDLE

First Party	Man in Middle	Second Party
P1=11	P1=11	P1=11
P2=7	P2=7	P2=7
X=3	X=8	Y=6
	Y=9	
$A=P2^X$ MOD P1	$A=P2^X$ MOD P1 $B=P2^Y$ MOD P1	$B=P2^Y$ MOD P1
A=2,8*	A=9,2* B=8,4*	B=4,9*
$K1=B^X$ MOD P1		$K2=A^Y$ MOD P1
K1=6	$K1=B^X$ MOD P1 $K2=A^Y$ MOD P1	K2=9
	$K1=2^9$ MOD 11=6 $K1=4^9$ MOD 11=9	

All the three will calculate their keys as published in the table. We notice that the first party calculate the only K1, the second party calculates K2 whereas a man in middle calculates both K1 and K2.man in middle needs two keys .this is because at one side it is communicating with the first party and on other side communicating with the second party only then he can manipulate the data. The man-in-the-middle attack can work in opposition to the Diffie-Hellman key-exchange algorithm, resulting it to fail. This is due to the man-in-middle create actual communications believe that they are talking to each other, whereas in reality talking to man-in-middle.

B. Elliptic Curve Diffie-Hellman Key Exchange Algorithm

It is a key negotiation protocol between two entities each consists of an elliptic-curve public-private key pair, to create a mutual secret on an insecure channel. This mutual secret might be used as a key, or to derive an additional key. The key, or the consequent key, can then be used to encrypt succeeding interactions using a symmetric-key cipher. It is a modification of the Diffie–Hellman scheme using elliptic-curve. For creating a mutual secret in between two parties A and B using Elliptical curve cryptography, both must have to make an agreement on Elliptical curve domain constraint. Both parties must contain a key pair off including a private key d (d is an integer value less than n, where n is the order of curve) and other public key Q=d*G where G is generator polymonial. Let (dA, QA) be the private-public key pair of A and (dB, QB) be the private-public key of B. at the end A computes KA=(XA, YA)=dA*QB. And B computes KB= (XB,YB)-dB*QA. Since dA*QB=dAdB G=dBdA G=dB*QA. therefore KA=KB and hence XA=XB. The mutual secret is KA since it is basically unfeasible to deduce the private key dA or dB from the public key KA.

C. Asymmetric Key Exchange

In this method, the communicating parties e.g. A and B don't approach trusted server T for lock and key pair at same time, instead only single party B approach T acquire a lock and key K1 that stamp the lock and transmit the lock along with key K1 to A. B communicate with A to use that lock and key to seal the packet prior to sending to B. now how B can unlock the lock, B request a different Key K2 from T only which can unlock the packet. Even the key K1 which is used for locking cannot unlock the packet. This technique is known as asymmetric operation. T is trusted third party. it means B possesses a key pair .one key K1 is used for locking and only equivalent other key K2 from the pair can be used for unlocking. Here K1 is public key whereas K2 is known as private key, the entity B can send the lock and the key K1 to anybody who wants to send message securely to B. Consequently B can Unlock the message by using his same private key K2.if B want to receive message securely from thousand sender even then B can share same lock and public key K1 and same private key K2 used for open the lock by the receiver that is B. it is one of the best methods, as compared to symmetric key operation.

VI. KEY RANGE AND KEY SIZE

Even highly secured encryption algorithm can be broken if a person has knowledge about the key size and key range [22]. The quantity of bits in key used by a cryptographic algorithm defines the key size. Key length describes the upper-bound on an algorithms security .most symmetric-key algorithms are designed to have security equal to their key length. After describing the key size and key length new attack might be revealed. As long as the relationship between key length and security is adequate for an application, then it does not matter if key length and security overlap. This is an important concept for the asymmetric-key algorithm. Elliptic curve cryptography plays an important role where half of their key length provide effective security. The key range is a total number of keys from smallest to largest available keys. If an attacker has knowledge of the cryptographic algorithm and about the encrypted message, there is only key-value remain challenge. So in order to find the right key, all the values within a key range can be tried using brute force attack until the right key not found. Studies have found that on average, the right key is found after trying half of the possible keys in the key range. and the concept of key-range show the way to the method of key size. Therefore increasing the key range to a large extent will leads longer time find right key using brute force attack. Key size can be measured in bits and is represented using a binary number system. Thus if the key range from 0 to 8, the key size is 3 bits. key size may vary,

depending upon the applications and the cryptographic algorithm being used, it can be 40 bits,64 bits, 128 bits and so on. In practice, 40-bit key takes about 3 hours to crack, 42 bits takes 12 hours and so on. It concludes that every extra bit double the time required to crack the key. Now a day's 128-bit key is quite safe, taking into consideration today's computers processing power.

VII. KEY RING

It is a data structure that contains the public and private key. It contains the public key to communicating users and some certificates and related information. There are two types of key ring [20].

A. Private key ring

The private Key ring includes the following fields:

Time Stamp: this field contains the time of generation of the key pair.

Key ID: this tells the identity of the public key.

Public key: this contains the public key of the user.

Private Key: this field contains the encrypted key of the user.

User ID: this field contains the identification number of communicating user.

TABLE III. PRIVATE KEY RING

Time Stamp	Key ID	Public Key	Encrypted Private Key	User ID
Time Stamp	Key ID	Public Key	Encrypted Private Key	User ID
---	--	--	--	--
--	--	--	--	--
Time Stamp	Key ID	Public Key	Encrypted Private Key	User ID
---	--	--	--	--
--	--	--	--	--

B.Public Key Ring

The public key ring is used to obtain the public key of other users. It contains the following fields.

TABLE IV. PUBLIC KEY RING

Time Stamp	Key ID	Public Key	Owner Trust	User ID	Key Legitimacy	Signature	Signature Trust
Time Stamp	Key ID	Public Key	Owner Trust	User ID	Key Legitimacy	Signature	Signature Trust
--	--	--	--	--	--	--	--
Time Stamp	Key ID	Public Key	Owner Trust	User ID	Key Legitimacy	Signature	Signature Trust
--	--	--	--	--	--	--	--

Owner trust*:* this field point toward the scale of trust to which level this public key is used to sign another public key certificate. This degree of trust is assigned by the user.

Key legitimacy: this field point toward the degree to which level this public key is trusted.

Signature trust: this field points out the degree to which user trust the signer to endorse the public key.

VIII. RANDOM NUMBER GENERATION

Randomness is a property that is strongly related to security in cryptographic system [10]. The random number carry the secrete but does not give any clue about the information. Randomness in created by probabilistic processes that generate homogeneously scattered and unpredictable numbers that can never be easily reproduced. In cryptography there are different techniques to produce random number sequence. Based on the properties of random number generators are classified in to three types:

A. True Number generator

True random number generation depends on physical properties of the source like photoelectric effect, temperature effect, vibrations, atmospheric noise and others. True random number generation can also be based on operators input properties like typing on the keyboard, mouse movement. True number can be generated by a physical as well as non physical source. The physical true random number generator is a noisy device like electronic circuits e.g. Diodes and oscillators, it may also be quantum effect of photons, analog to digital signals converter. The physical true random number generators are a noisy device like electronic circuits e.g. Diodes and oscillators, it may also be quantum effect of photons, analog to digital signals converter. The digitized analog signal algorithmically processed in order to decrease any weakness. The random bits can be generated using deterministic algorithm where random number is generated from a seed value. These devices also known as deterministic random bit generators. The person who does not have any knowledge about the used seed value cannot predict the output.

B. Unpredictable Random Number Generator

Unpredictable random numbers extort randomness from the surrounding devices like computer component but awareness of the underlying process may simplify the calculation of internal states and hence next value. They are based on the activities of hardware devices like true random number generators, yet unpredictable random number generator perform a deterministic sequence of operation. The effect of the large amount of events and factor is so complex and any modification in the production method is impractical for an adversary to calculate the produce output. Unpredictable random number generators extort the unpredictability produced by the complication of the core process, such as human computer interaction, race condition and internal processor states.

C. Pseudorandom Number Generator

The source for randomness is initial value known as seed, which is extended by a deterministic procedure providing the random number creation sequence using software methods. The output is totally dependent on starting state of generator and as a result by the seed.PRNGS such as high production speed, excellent statistical results and no need for extra hardware make PRNGS most widely used random number generators in cryptography. But it doesn't show sufficient strength and can be easily compromise by many

types of attacks. In order to increase its strength many new technique like elliptical curve, integer factorization and use of hash function are used.

IX. CRYPTOGRAPHIC HASH FUNCTION

A cryptographic hash function is a group of a hash function that has definite characteristics which formulate appropriate for use in cryptography. It is a algebraic equation that map data of random size to a bit string of predetermined size and is calculated to be a one-way function. This type of function is impossible to invert. The perfect cryptographic hash function has the characteristics like it must be fast to calculate the hash value for any specified message, infeasible to produce the original message from its hash value, impossible to find two separate messages with identical hash value. Cryptographic hash function have many application such as digital signature, message authentication system, index in the hash table, fingerprint etc.

Cryptographic hash functions provide the security based on nature of application in which they are used:

a. It provides message integrity when applied in pre-processing steps for a digital signature algorithm.

b. It provides information integrity when it is attached with the encrypted message.

c. It provides redundancy when appended to data before encryption.

d. It protects the password

e. It is used for pseudo-random string production or for key derivation in general application.

f. It provides basis for creation of MACs, stream ciphers and block ciphers.

The National Institute of standard and Technology (NIST) developed a set of algorithm known as a hash algorithm. These secure encryption function developed to meet cyber security challenges.

A hash function h has few properties like.

h maps an input x of finite bit piece to an output h(x) of preset bit length n.

$h:\{0,1\}^* \rightarrow \{0,1\}^n$

if h and x are known then it is easy to compute h(x).

if y is the output then it must be computationally infeasible to compute x such that h(x)=y.

e.g. $h(x) = g^x \bmod p$ where is a prime number and g is a primitive root in Z^*_p.

Here h is a one-way hash function.

A cryptographic hash function h assures some additional properties like.

- h is one-way function, for every pre-determined outputs y,it is difficult compute an x such that h(x)=y.
- given x it is infeasible to find second pre-image x' with x=x' such that h(x) =h(x').
- it is computationally impossible to distinguish h(m) from random n-bit value.
- h should acknowledge a block of data of any volume as input
- h should generate a fixed-length output not dependent on length of input block
- h should acts like a random method while being deterministic and competently reproducible.
- h must be calculated in polynomial time O(n) where n is the size of input message that makes hardware as well as software implementation easy

There are four category of secure hash algorithm SHA-1,SHA-256, SHA-384 and SHA-512.these are iterative, one-way hash formulas that can compute a message to generate a condensed demonstration called a message digest. Each of these algorithms can be expanded in two stages: pre-processing and hash calculation. The hashing process produce a message plan from the padded message and utilize that schedule, along with function constant to iteratively create a sequence of hash values. The concluding hash value engendered by the hash calculation is used to decide the message digest. These four algorithms vary drastically in the quantity of bits of security that are used for data being hashed – this is straightforwardly associated to the message digest size. When a secure hash algorithm is used in combination with another algorithm, there may be necessities specified somewhere else that need the use of a secure hash algorithm with particular number of bits of security. For example, if a message is being associated with a digital signature algorithm that present 128 bits of security, then that signature algorithm may need the utilization of a secure hash algorithm that also presents 128 bits of security (e.g., SHA-256). Moreover, these four procedures be different in terms of the dimension of the blocks and words of data that are used for the duration of hashing process. Figure 1 presents the fundamental characteristics of all four safe hash algorithms.

A. Message Digest-2

Message Digest 2 (MD2) is a hash function designed by Ron Rivest 1989.it produces a 128-bit hash value by using an random length message.MD2 was developed for digital signature applications. it is used in public key infrastructure but rarely used as it takes more computational time and considered to be no longer secure.MD2 has withstood cryptanalytic attacks for a long time. the MD2 varies from its successors MD4, MD5, and SHA-1 by using S-box. The first assault against the MD2 hash function was a pre-image attack in the year 2004 published by F.Muller [16].

B. SHA-1 Functions

SHA-1 make use of a series of logical formulas f_0, f_1------ f_{79}.every formula f_t, where 0<=t<=79, work on three 32-bit words, x, y, z and produce a 32-bit word as output. The function f_t (x,y,z) is defined as follows.

$$F_t(x,y,z) = \begin{cases} Ch(x,y,z) = (x \wedge y) \oplus (x \wedge z) & 0 \le t \le 19 \\ Parity(x,y,z) = x \oplus y \oplus z & 20 \le t \le 39 \\ Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) & 40 \le t \le 59 \\ Parity(x,y,z) = x \oplus y \oplus z & 60 \le t \le 79 \end{cases}$$

Both the choice (Ch) and majority (Maj) functions work on individual bits and are balanced on their particular input domain.

C. SHA-256 Functions

In SHA-256 there are operations of six logical functions, where each function work on 32-bit words, which are

characterize as *x*, *y*, and *z*. The outcome of every function is a new 32-bit word.

$$Ch(x,y,z) = (x \wedge y) \oplus (x \wedge z)$$

$$Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$

$$\sum_0^{256}(x) = ROTR^2(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x)$$

$$\sum_1^{256}(x) = ROTR^6(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x)$$

$$\sigma_0^{256}(x) = ROTR^7(x) \oplus ROTR^{18}(x) \oplus SHR^3(x)$$

$$\sigma_1^{256}(x) = ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x)$$

The important characteristics of the Σ_0 and Σ_1 operator involved in the state register renew function and of the σ_0 and σ_1 involved in the message schedule calculations.

The circular right shift also know as rotate right operation $ROTR^n(x)$, where *x* is a *w*-bit word and *n* is an integer with $0 < n < w$, is characterized by

$$ROTR^n(x) = (x >> n) \text{ v } (x << w - n).$$

Thus circular shift of x by n positions to the right is equivalent to $ROTR^n(x)$.

The message block size for SHA-1 and SHA-256 is 512 bits Which are symbolized as a string of sixteen 32-bit words.

D. SHA-384 and SHA-512 Functions

Each message block has 1024 bits for SHA-384 and SHA-512, which are symbolized as a series of sixteen 64-bit words. Which are symbolized as *x*, *y*, and *z*. The result of each function is a new 64-bit word.

$$Ch(x,y,z) = (x \wedge y) \oplus (x \wedge z)$$

$$Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$

$$\sum_0^{512}(x) = ROTR^{28}(x) \oplus ROTR^{34}(x) \oplus ROTR^{39}(x)$$

$$\sum_1^{512}(x) = ROTR^{14}(x) \oplus ROTR^{18}(x) \oplus ROTR^{41}(x)$$

$$\sigma_0^{512}(x) = ROTR^1(x) \oplus ROTR^8(x) \oplus SHR^7(x)$$

$$\sigma_1^{512}(x) = ROTR^{19}(x) \oplus ROTR^{61}(x) \oplus SHR^6(x)$$

The (W_t) message schedule recurrence is shown in Figure. It can easily presented by a 16-stages feedback register [21].

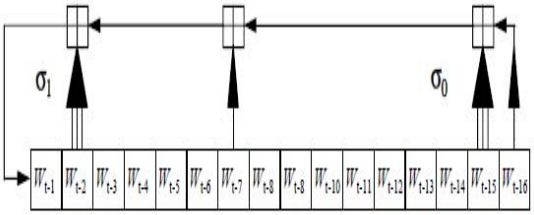

Fig 2: Message Schedule Recurrence

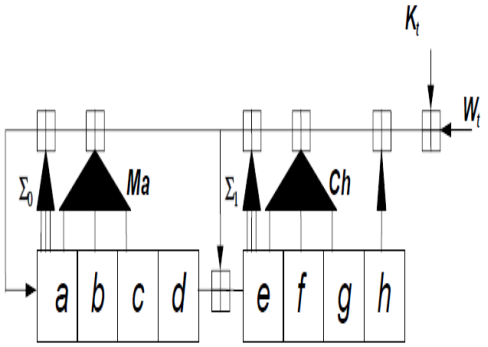

Fig 3: Hash Computation, State Register Update Function

the SHA-1 message sequence calculation, the SHA-384/512 message sequence calculation are not GF (2)-linear, because of the contribution of the + addition instead of XOR operation ⊕.this complicates the message sequence recurrence and is more difficult to analyze, since the set of potential differentiation patterns is no longer a linear code[21].

In SHA-1, the recurrence relation combines the variety of bit location is made stronger, due to the association of the bit rotation.

TABLE V. COMPARATIVE ANALYSIS OF ENCRYPTION ALGORITHMS

Algorithm	No of keys	Algorithm used	Output size (bits)	Speed of algorithm	Effect of key compromise	Complexity	Key management and sharing	Internal state size (bits)	Block size (bits)	Rounds	Operations	Security (in bits) against collision attacks	First Published
MD2	0	MD	128	fast	NA	Medium	NA	128 (3 X32)	512	64	And, Xor, Rot, Add (mod 232), Or	≤18	1992
SHA-0	0	SHA	160	fast	NA	Medium	NA	160 (5 × 32	512	80	And, Xor, Rot, Add (mod 232), Or	<34	1993
SHA-1	0	SHA	160	fast	NA	Medium	NA	160 (5 × 32	512	80	And, Xor, Rot, Add (mod 232), Or	<63	1995
SHA-224 SHA-256	0	SHA	224 256	fast	NA	Medium	NA	256 (8 × 32)	512	64	And, Xor, Rot, Add (mod 232), Or, Shr	112 128	2004 2001
SHA-384 SHA-512	0	SHA	384 512	fast	NA	Medium	NA	512 (8 × 64)	1024	80	And, Xor, Rot, Add (mod 264), Or, Shr	192 256	2001
SHA-512/224 SHA-512/256	0	SHA	224 256	fast	NA	Medium	NA	512 (8 × 64)	1024	80	And, Xor, Rot, Add (mod 264), Or, Shr	112 128	2012
SHA3-224 SHA3-256 SHA3-384 SHA3-512	0	SHA	224 256 384 512	fast	NA	Medium	NA	1600 (5 × 5 × 64)	1152 1088 832 576	24	And, Xor, Rot, Not	112 128 192 256	2015
Symmetric key algorithm	1	AES	128	fast	Loss of both sender and receiver	Medium	complex	512 (8 × 64)	128	10	XOR,ROTL,SHL	128	2001
Asymmetric key algorithm	2	RSA	2048	Relatively slow	Only loss for owner of Asymmetric key	High	Easy and secure	512 (8 × 64)	128	10	XOR,NOT	128	1977

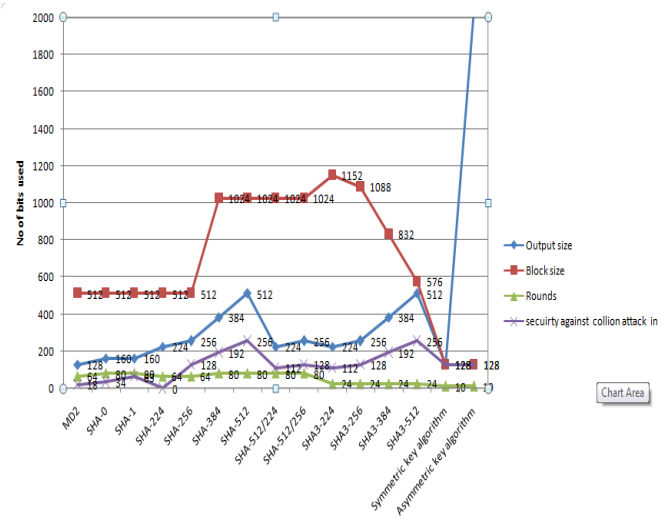

Fig 4. No of bit used vs encryption parameters

TABLE VI. OPERATOR TRADEOFF IN ENCRYPTION ALGORITHMS IN THE ALGORITHM DESIGN THERE IS TRADEOFF IN BETWEEN OPTIMIZATION OF HARDWARE DESIGN AND SOFTWARE DESIGN

Sr.No	Addition/Rotation/XOR	And/Rotation/XOR
1	In these operations there is efficient use of CPU instructions.	It is less software oriented
2	These operations are not good in hardware design.	These operations are good in hardware
3	These operations are hard to study and implement.	Comparatively easier to study and implement.

X. KEY GENERATION USING ELLIPTICAL CURVE CRYPTOGRAPHY

The elliptical curve cryptography depends on the algebraic characteristics of elliptical curves whereas the RSA depends on the hypothesis, that it's hard factoring of huge numbers where factors are two big prime numbers. the key generation using elliptical curve start with a correctly selected elliptical curve E represented over a finite field Fq of characteristics p,and base point P € E(Fq) [2]. On elliptical curve for given two points P(Xp,Yp), Q(Xq,Yq) (P≠±Q) in the group E.The group operator will permit us to evaluate a third point R(Xr,Yr), also in the set E, such that P + Q = R.The coordinates of point R: $Xr = S^2 - Xp - Xq$ where $S^2 = 2Xp + Xq + Xr - Xp$. As point R fit in to the straight line (PQ) then S= Yr−Yp/Xr−Xp and we find: Yr = Yp +S(Xr –Xp)

The domain parameter in elliptical curve consists of:

a. A field with size q, where q is a prime power (q=p, odd prime, $q=2^m$).

b. The notation FR known as field demonstration used for elements of Fq.

c. The field values a and b in Fq defines the expression of the elliptical curve E over Fq.

d. The finite point P= (xp, yp) of prime order in E (Fq).since P is defined by two field elements xp, yp, this indicates that P does not point at infinity.

e. The sequence of point P is represented by n.

f. the cofactor h=E(Fq)/n

With given set of domain factor (q,FR,a,b,P,n,h) ,the private key is an integral value d elected randomely from the set [1 to n-1].and public key is the point on elliptical curve Q =dP. The key pair is (Q,d).and each exectution cycle of a key negotiation protocol in between two entities A and B should generate a unique common secret key.

The elliptical curve use scalar multiplication operation in all applications. In this operation large integer is multiplied by a point by using a series of point doubling and addition until the final product vale is reached on the curve [26].

INPUT: An integral value k>0 and a point value P.
OUTPUT: Q=k*P
1. Set k← (kl-1…k1k0)2
2. Set P1←P, P2=2P.
3. for I from I-2 until 0 do
If k_i=1 then
Set P1←P1 +P2, P2←2P.
Else
Set P2←P2+P1, P1←2P1.
4. RETURN (Q=P1)

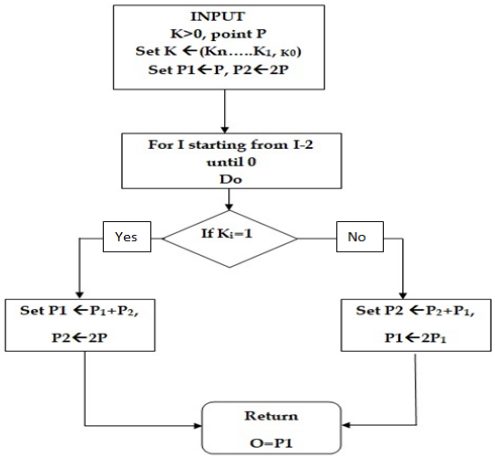

Fig 5: ECC algorithm flow chart

The computing time required to solve discrete logarithms in cyclic subgroups of E (Fp) with various order n by Koblitz, Menezes and Vanstone is summarized in table given below [18].

TABLE VII. NUMBER OF BITS USED ENCRYPTION KEY VS. TIME REQUIRED TO DECRYPT THE ALGORITHM

Size of n (bits)	MIPS(years)
512	$3x\ 10^4$
768	2×10^8
1024	2×10^{11}
1280	$1x10^{14}$
1536	3×10^{16}
2048	3×10^{20}

XI. COMPARISON OF KEY GENERATION

The elliptic curve cryptographic system significantly reduces the size of the encryption keys in comparison to discrete logarithms which relatively took high computing resources. The short key size facilitates more rapidly execution of different cryptographic operations. So it is concluded that RSA key generation take more time in comparison to the elliptic curve based crypto system of equivalent level of protection [19].

TABLE VIII.: KEY SIZE VS. KEY GENERATION TIME

Key range (bits)		Generation Time (seconds)	
ECC	*RSA*	*ECC*	*RSA*
163	1024	0.08	0.16
233	2240	0.18	7.47
283	3072	0.27	9.89
409	7680	0.64	133.90
571	15360	1.44	679.06

TABLE IX. KEY SIZE FOR EQUIVALENT SECURITY [23]

Symmetric Scheme (key size in bits)	ECC-based Scheme (size of *n* in bits)	RSA/DSA (modulus in bits)
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

From above tables, it is clear that, due to short key size the key generation time in elliptic curve based crypto systems is considerably faster than RSA. Also level of security considerably increases the generation time ratio.

XII. CONCLUSION

In spite of decades back history of cryptography there is deficiency of research in encryption techniques for wireless sensor networks. The traditional algorithm like RSA, AES are extensively studied and widely used in providing security over the internet. But lack of research in complex mathematical foundations of elliptic curves fails to provide security solutions for resource limited system solutions such as sensors, smart cards, and mobile devices. There are implementation constraints in a secure cryptosystem related to usage of defective random number generators, cryptographic hash functions and inadequate security to the key distribution system. In future, it is expected that elliptic curves play a significant role in various implementations of lightweight encryption algorithms due to a small key size and fast encryption and decryption operation.

REFERENCES

[1] Ram Ratan Ahirwal et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (2) , 2013, 363 – 368.

[2] Dragan Vidakovic et. al."Generating Keys in Elliptic Curve Cryptosystems "International Journal of Computer Science and Business Informatics ISSN: 1694-2108 | Vol. 4, No. 1. AUGUST 2013.

[3] Ansah Jeelani Zargart et al, International Journal of Advanced Research in Computer Science, 8 (7), July-August 2017, 48-51.

[4] Krists Magons,"Applications and Benefits of Elliptic Curve Cryptography" University of Latvia, Faculty of Computing, Rain,a bulvaris 19, Riga, LV-1586, Latvia.

[5] Rajeev Sobti,"Cryptographic Hash Functions: A Review",IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 2, March 2012.

[6] Dr. S. Vasundhara,"The Advantages of Elliptic Curve Cryptography for Security "Global Journal of Pure and Applied Mathematics.ISSN 0973-1768 Volume 13, Number 9 (2017), pp. 4995-5011.

[7] Madhumita Panda,"Security in Wireless Sensor Networks using Cryptographic Techniques"American Journal of Engineering Research (AJER) e-ISSN : 2320-0847 p-ISSN : 2320-0936 Volume-03, Issue-01, pp-50-56.2012.

[8] Mansoor Ebrahim,"Symmetric Algorithm Survey: A Comparative Analysis "International Journal of Computer Applications (0975 – 8887) Volume 61– No.20, January 2013.

[9] Kinga MÁRTON et.al. "Generation and testing of random numbers for cryptographic applications "proceedings of the Romanian academy, Series A, Volume 13, Number 4/2012, pp. 368–377

[10] Kinga MARTON et. al. "Randomness in Digital Cryptography: A Survey "Romanian journal of information science and technology Volume 13.

[11] M.R.Faghani,S. M. Motahari,: Sectorized Location Dependent Key Management",IEEE International Conference on Wireless and Mobile Computing, Networking and Communications 12-14 Oct (2009)

[12] Blackburn S.R., Etzion T., Martin K.M., Paterson M.B.: Efficient Key Predistribution for Grid-Based Wireless Sensor Networks. In: Safavi-Naini R. (eds) Information Theoretic Security. ICITS 2008. Lecture Notes in Computer Science, vol.5155. Springer, Berlin, Heidelberg. (2008)

[13] Mu Kun and Li Li: An Efficient Pair wise Key Predistribution Scheme for Wireless Sensor Networks. Journal Of Networks,9(2)(2014)

[14] Abdoulaye Diop, Yue Qi, Qin Wang, and Shariq Hussain: An Efficient and Secure Key Management Scheme for Hierarchical Wireless Sensor Networks.International Journal of Computer and Communication Engineering.1(4)(2012)

[15] G. Gaubatz, J.-P. Kaps, and B. Sunar(2004). 'Public key cryptography in sensor networks revisited in 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004).

[16] F. Muller. The MD2 Hash Function Is Not One-Way. In P. J. Lee, editor, Advances in Cryptology – ASIACRYPT 2004, Proceedings, volume 3329 of Lecture Notes in Computer Science, pages 214–229. Springer, 2004.

[17] Vishal Choudhary,S.Taruna: Improved Key Distribution and Management in Wireless Sensor Network. Journal of Wireless Communications 1 (1): 16-22, (2016). Retrieved from www.sispress.org

[18] Koblitz, N., Menezes, A., Vanstone, S.: The State of Elliptic Curve Cryptography. In:"Towards a Quarter-Century of Public Key Cryptography", Kluwer Academic Publishers,pp. 173–193, Boston (2000).

[19] Arrendondo, B., Jansma,N. "Performance Comparison of Elliptic Curve and RSA Digital Signatures". IPCSIT vol. 4, IACSIT Press, Singapore (2011).

[20] Dr. Manoj kumar,"Cryptography and network Security" Krishna Prakashan media(P) Ltd.ISBN:81-8283-027-3(2008)

[21] Dr Helena,Henri Gilbert, "Evaluation Report Security Level of cryptography "France Telecom R&D ,Issy-les-Moulineaux (2002).

[22] Serge Vaudenay,"A classical introduction to modern cryptography Applications for Communications Security" Swiss Federal Institute of Technologies (EPFL),Springer Science Business Media, Inc.(2006)

[23] William Stalling, "Cryptography and network security principles and practice "Pearson Education, Inc., publishing as Prentice Hall (2006).

[24] Atual Kahate,"A text book on Cryptography and Network Security"TMH, fourth edition (2008).

[25] Dongmei Jiang Dafu Lou. "Personalized Service Mobility and Security in SIP-based communications", 2005 13th IEEE International Conference on Networks Jointly held with the 2005 IEEE 7th Malaysia International Conf on Communication, 2005.

[26] López J., Dahab R. Fast Multiplication on Elliptic Curves Over *GF*(2m) without precomputation. In: Koç Ç.K., Paar C. (eds) Cryptographic Hardware and Embedded Systems. CHES 1999. Lecture Notes in Computer Science, vol 1717. Springer, Berlin, Heidelberg.

[27] Krawczyk H. Cryptographic Extraction and Key Derivation: The HKDF Scheme. In: Rabin T. (eds) Advances in Cryptology – CRYPTO 2010. CRYPTO 2010. Lecture Notes in Computer Science, vol 6223. Springer, Berlin, Heidelberg.

[28] Zhen-Rong Zhu, Hui Li, Yi-Xian Yang. "Efficient Anonymous Sever-Supported Signature Protocol Based on Multipath Hash Chain", 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008.