An Enhanced Elliptic Curve Cryptosystem for Securing Data

Edward Kwadwo Boahen
Kwame Nkrumah University of Science and Technology
Department of Computer Science

James Ben Hayfron-Acquah
Kwame Nkrumah University of Science and Technology
Department of Computer Science

Frimpong Twum, PhD
Kwame Nkrumah University of Science and Technology
Department of Computer Science

ABSTRACT
The purpose of this research is to enhance the cryptographic system called the Elliptic Curve. Elliptic Curve cryptosystem (ECC) is a technique of public-key encryption, which is rooted on the arithmetical construction of elliptic curves over finite fields. Elliptic Curve Cryptographic System necessitates smaller keys compared to non-ECC cryptography to offer equal security. The security of RSA is based on the computational task of considering extensive numbers leading to an increase in encryption computation time, slower connection of the SSL handshake and increase in CPU usage during handshakes. Therefore, there should be a new way of solving this problem, which is ECC encryption. Elliptic curves are effective for digital signatures, key agreement, generators, pseudo-random and other related tasks. The first phase of the project involves understanding the key exchange of Diffie-Hellman and applying the properties of the Elliptic Curves. It is terminated with key facts that the Elliptic Curve Cryptography has a shorter key length, saves bandwidth, which facilitates key generation during the encryption/decryption of data, also the assurance of faster encryption and decryption, and notwithstanding its efficiency and efficacy in small devices.

General Terms
Elliptic Curve Cryptography; Diffie-Hellman; Secured Socket Layer; encryption; decryption; Rivest-Shamir-Adleman; Hypertext transfer protocol; Triple DES; Digital Signature Algorithm., Algorithms et. al.

Keywords
Elliptic Curve Cryptography; Diffie-Hellman; Secured Socket Layer; encryption; decryption; Rivest-Shamir-Adleman; Hypertext transfer protocol; Triple DES; Digital Signature Algorithm.

1. INTRODUCTION
Cryptography is the technique that when implemented can be used to secure data on a network. It is also the science of securing information that is transmitted in a manner that the authorized recipient only is able to retrieve or access the information. As far as communication is concern, it is vital to encrypt the message to conceal it from intruders. Network security is primarily based on cryptography (Stallings, Data and Computer Communications, 2005). The art of protecting information (encryption) is usually done by converting it into associate undecipherable format (encrypted text), which is referred to as the cipher text.

2. BACKGROUND
The objective of secure communications is based on confidentiality or secrecy (S), that is, to cover the contents of a publicly exposed message from unauthorized recipients or intruders. In modern-day business and communications, it is normally difficult for the receiver to know and confirm if the communication has not been tempered with at some stage during the transmission or whether an intruder has tampered with it. One major problem in message authentication is that, you cannot tell exactly if it is the actual message since there is usually no delay in transmission. In this research, HTTP communications securities are discussed with emphasis on applications that cannot be satisfactorily threatened by means of current cryptographic procedures. A completely new idea, the elliptic curve encryption/decryption channel solves the brand-new requirements in comfy transmissions. Cryptosystems are symmetric when the sender and recipient secretly possess the same key. The secret keys mentioned are used within the encoding procedure to present indecision, which can be detached in the procedure of decryption by the certified receiver using his key. With this method if the key currently used by both parties is compromised, the communication between the two parties would not be secure and hence will create a barrier. The new cryptosystems are asymmetric in the sense that there are two keys being used by the parties in the communication. One of the keys is public to both and the other is private. Secure HTTP Communication is a prerequisite for today's online exchanges and communication. Whether trading financial, business or individual data, individuals need to be in the know of the parties involved in the conveying (verification) as well as the surety that the records will not be reformed (information trustworthiness) nor unveiled (confidentiality) during transmission. As such, the Secure Sockets Layer (SSL) protocol is considered as one of the best alternatives in attaining these objectives. The SSL convention is application free – theoretically, applications that keeps running over TCP will likewise keep running over SSL. Making it an imperative motivation why the spoken about transmissions have outperformed that of additional security conventions, for example S/MIME, SET and SSH, and. There are a number of cases of utilization conventions like LDAP, FTP, TELNET and IMAP and using Secure Shell Layer in various transmissions. In any case, the greatest use of Secure Shell Layer is for safeguarding Hypertext Transfer Protocol, the principle convention regarding the World Wide Web (Frier, Karlton, & Kocher).

Amid its origination with Netscape in 1990s, throughout its institutionalization inside the IETF (Internet Engineering Task Force) around the later years of the 20th century, the convention and its usage have been examined by a distinguished security expert in the world. Currently, the Secure Shell Layer is trusted to protect exchanges for delicate systems going from financial transactions, to stock exchange, to electronic business. The utilization of Secure Shell Layer forces a prominent execution price on web servers. Coarfa et al. (2006) having testified that web servers that are secured

runs 3.4 to nine times slower contrasted with steady web servers on similar equipment stage. Moderate reaction time is a noteworthy reason for disappointment for online customers and regularly drives them to surrender their electrical spending baskets amid look-up time frustration in waiting. As indicated by one gauge, the potential income misfortune from e-trade exchanges prematurely ended because of Web execution issues surpasses a few billion dollars every year (Wagner & Schneier, 1996).

SSL uses RSA mode of encoding to convey an arbitrarily picked mystery that is utilized to determine keys for information encoding and confirmation. The RSA decoding procedure is the most figure concentrated section of the Secure Shell Layer exchange for a safe web server. A few merchants, for example Rainbow, nCipher, Sun and Broadcom and now offer specific equipment to devolve RSA calculations to improve its server execution. This research sightsees the use of Elliptic Curve Cryptography (ECC), a resourceful option to RSA, as a means of refining the Secure Shell Layer performance without selecting posh superior purposeful hardware. ECC was first projected by Victor Miller and singularly by Neal Koblitz in the middle of the 20th century and has changed into an established public-key cryptosystem.

3. PROBLEM STATEMENT

It is evident that the security of RSA is based on the computational task of considering extensive numbers. As such, as figuring power increases, and calculations that are more productive are established, the capacity to consider lengthier numbers increases. The quality of encryption by directly fixing to key size and multiplying key length conveys an exponential increment in quality, though it is considered to it impair execution. Basically, RSA keys are either 1024-bits or 2048-bits in length. With the increase in computing power, specialists trust that the 1024-piece keys could be eased up soon. That is why stakeholder is looking into the implementation of a base key length of 2048-bits.

Because the strength of the security in the HTTP-SSL transmission is dependent on the encryption key size, expects try to increase the key size to 4096 bit. The increase in the key size therefore requires the technical staff to face the following issues:

A. There is an increase in encryption computation time.
B. The SSL handshake at the start of each connection would be slower.
C. There is an increase in CPU usage during handshakes.

Websites that have busy data exchange such as amazon.com, google.com, and the likes cannot afford to defray customers with such connectivity issues and the problems stated up here. Therefore, there should be a new way of solving these problems that is why ECC encryption is gaining popularity and attention.

4. RELATED WORK

Cryptography has undoubtedly become one of the most principal areas of communication security and thus an imperative block of computer security. Cryptographic techniques provide a secure communication in the occurrence of adversaries to stay abreast with data securities such as data authentication, non-repudiation, confidentiality, and information integrity.

The method of converting plain text or ordinary information into incomprehensible text classified as cipher text in cryptography is termed encryption. This cipher text is perceivable exclusively to someone who is aware of a way to decode it, and any resister that may understand the cipher text should not be intelligent to confirm all-round the first encrypted message. Furthermore, any lawful party can decipher the cipher text using a secret writing algorithmic rule that sometimes requires a secret writing key. With that well stated, there are two main frameworks used for providing the absent protection by converting a message into a state, where if it were captured in transfer, the innards of the novel message could not be overtly exposed, and these techniques fall under two generic classes and these are show in table 1 below.

Table 1. Encryption Algorithm Types

Encryption Algorithm Types	
Symmetric	**Asymmetric**
AES(AES-128, AES-192, AES-256)	DIFFIE-HELLMAN KEY EXCHANGE
BLOWFISH	RSA ASYMMETRIC ALGORITHM
TWOFISH	SHA-224
DES	SHA-256
3DES	SHA-386
RC4	SHA-512
	SHA-3(emerging standard)

4.1 Symmetric Encryption

Per symmetric encryption, the undistinguishable key is used to decrypt and encrypt the information or data as shown in Fig. 1. Symmetric key algorithms are considerably faster than asymmetric algorithms. This is because; the process of encryption is not as much as complicated as asymmetric algorithms.

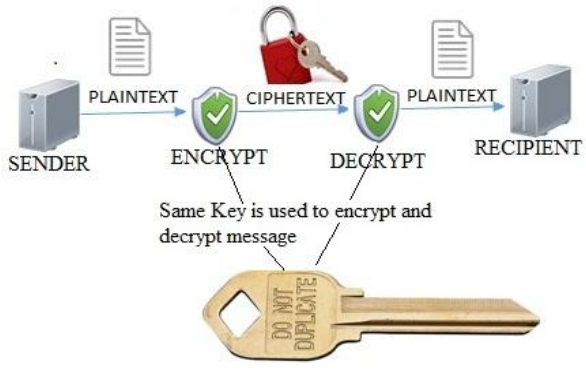

Fig 1: Diagram of Symmetric Encryption

4.2 Asymmetric Encryption

Public-key cryptography is as well-known as asymmetric encryption because of the spatial property in the control of key data by the parties involved. Specifically, one of the parties features a secret key whereas the other party has the universal public key that equals this secret key. This is often in distinction to the evenness within the personal key setting, wherever each party has an identical key.

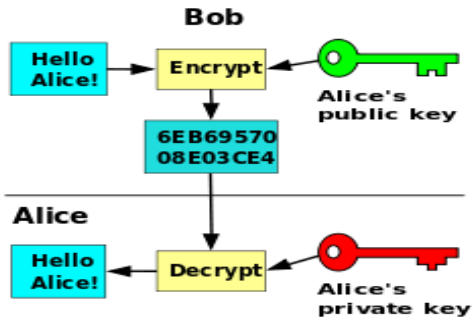

Fig 2: Example of Asymmetric Encryption

4.3 The Representation of Commonly Used Encryption Algorithms

The transport, altertion and generation, of keys are completed by encryption rules. It is conjointly termed as the cryptographic rule. There are numerous cryptographic algorithms available for the encryption of data and information. The forte of the encryption rule deeply depends on the computing device used for key generation. The encryption algorithms are vividly declared below.

4.3.1 Rivest-Shamir-Adleman (RSA)

Rivest et al. (1978) designed RSA, known to be amongst the finest recognized public key cryptosystems for key exchange or digital signatures or coding of blocks of knowledge. RSA utilizes an inconstant size-coding block and an adjustable key size. It is associate degree public key cryptosystem, buttressed variety theory, which may be a system for block cipher. It deploys two prime numbers to produce the universal public and private keys. The two very unalike keys are used for coding and decipherment purposes. The sender then encrypts the message with public key of the recipients which is then sent through a reliable, but not principally a secret route and once the message gets conveyed to the recipient, then the recipient will use his/her own personal key to decrypt the message. RSA operations are rotten in four (4) stages:

1. Key generation
2. Key distribution
3. Encryption
4. Decryption.

RSA has some disadvantages, and these are:

A. The Need of Security Proof. RSA is dependent on the drawback of the factorization of massive numbers but is identical to the factoring, which has not been verified hypothetically. In the mean time because there exists no evidence of broken or fragmented RSA factorization would be required. If there exists an algorithm that can rapidly crumble a huge number, the RSA algorithm's security would be jeopardized. Fig.3 demonstrates the arrangement of the algorithmic rule that RSA follows aimed at the coding of several blocks.

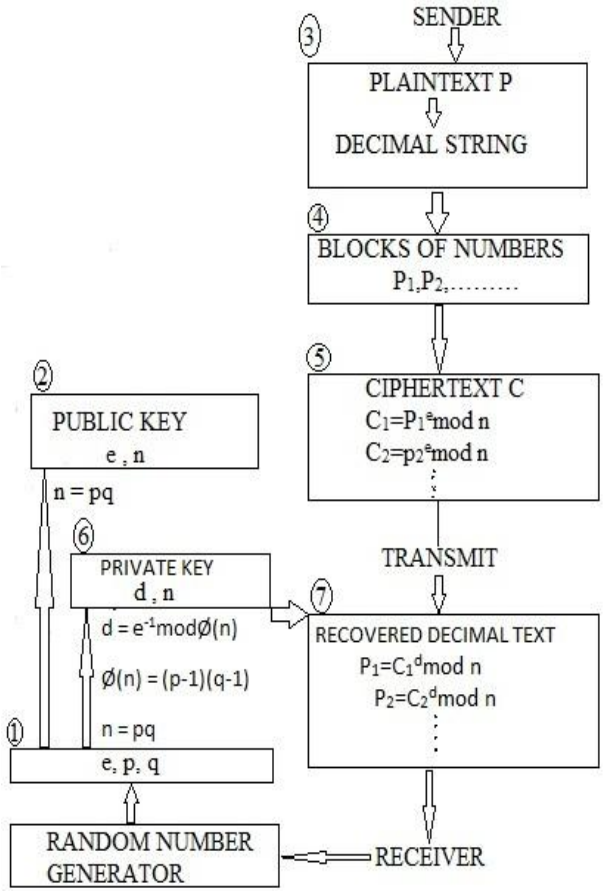

Fig 3: RSA processing of Multiple Blocks (Singh & Supriya, April 2013)

4.4 HTTP Secured Connection

Intrinsically, secured communication is highly demanded in this era of on-line transactions. In the exchange of information that may encompass the context of finance, trade or personal transactions, folks need to grasp with whom they are (authentication) and that they would like to confirm that the info is neither altered (data integrity) nor made known (confidentiality) in transit. The Secure Sockets Layer (SSL) protocol is considered the best alternative for achieving these goals. The SSL protocol is application freelance – abstractly, all applications that runs over TCP also can route over SSL. Thus making it a crucial reason why its preparation has outpaced that of alternative security protocols like SSH (Ylonen, Kivinen, Saarinen, Rinne, & Lehtinen, 2003) S/MIME and SET. The square measure several samples of application protocols such as TELNET, FTP, IMAP and LDAP which run evidently over SSL. Nevertheless, the prime purpose of SSL usage is to secure communication protocols (Fielding, 1999).

HTTPS constructs a protected path over an insecure network. This provides shielding from attacks like eavesdropping and man-in-the- middle attacks, given that sufficient cipher suites area unit is used of which the server certificate being used is verified and convinced. Since communications protocol piggybacks HTTP is wholly on high of the TLS, primarily everything about communications protocols are often encrypted.

Probably this may include the requested universal resource locator (which specific online page that has been requested), question parameters, the headers, and the cookies (which typically contain identity info regarding the user). However,

due to factors such as host (website) addresses and port numbers area units, that is an essential component of the TCP and IP protocols, thereby HTTPS cannot defend their revealing.

In observing this, advise is that even with a properly organized internet server, intruders who eavesdrops will deduce the IP addresses and also the port variety of the online server (occasionally even the name e.g. www.example.org, however not the remainder of the URL) that one is to act with, likewise because the quantity which is data conveyed and length which is the length of period of the communication, that is not the content of the communication.

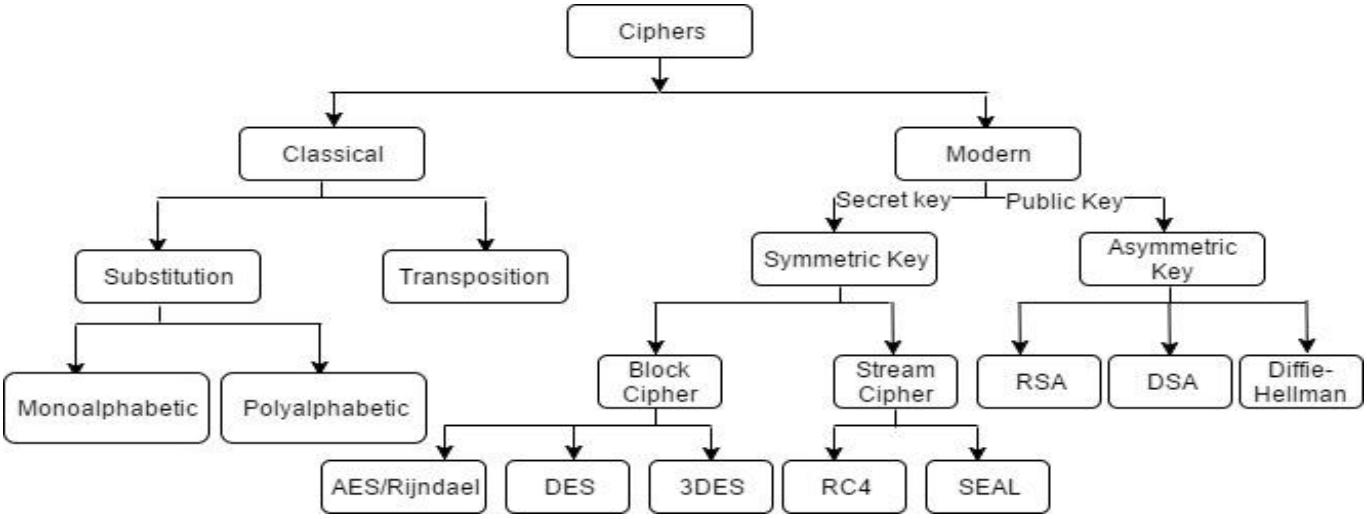

Fig 4: Example of Asymmetric Encryption

5. METHODOLOGY

Table 2 reviews the three kinds of popular public-key cryptosystems. The table carefully shows that DSA, Diffie-Hellman and RSA can all be vulnerable to attacks using sub-exponential algorithms, but the popular attack on ECC demands exponential time. Public-key systems are normally used to convey or share keys for symmetric-key ciphers. The work factor needed to crack, a symmetric key should match that needed to break the public-key system used during key exchange since the security of the systems is as robust as its feeblest constituent.

Table 2. A comparison of public-key cryptosystems (Vanstone, 2003)

Public-key system	**Examples**	**Mathematical Problem**	**Best known method for solving math problem(running time)**
Integer factorization	RSA, Rabin-Williams	Given a number n, find its prime factors	Number field sieve: $\exp[1.923(\log n)^{1/3}(\log\log n)^{2/3}]$ (Sub exponential)
Discrete logarithm	Diffie-Hellman(DH),DSA, ELGamal	Given a prime n, and numbers g and h, find x such that $h=g^x \bmod n$	Number field sieve: $\exp[1.923(\log n)^{1/3}(\log\log n)^{2/3}]$ (Sub exponential)
Elliptic curve discrete logarithm	ECDH, ECDSA	Given an elliptic curve E and points P and Q on E, find x such that Q=xP	Pollard-rho algorithm: $\sqrt{n}$ (Fully exponential)

In place of exposing the secret agreement directly, the EC Diffie-HellmanCng class performs a number of post-processing on the agreement before the value is provided. The post-processing technique also known as the key derivation function; here, you are able to choose which KDF you wish to implement and set its parameters via a set of properties on the instance of the Diffie-Hellman object

5.1 Proposed Procedure

The steps involved in the proposed procedure involves standard techniques such as the Diffie-Hellman key exchange algorithm, elliptic curve properties and computations, and mathematical principles in modular arithmetic and the Euclidean algorithm, mathematics of cryptography.

5.2 The Steps Involved in the Elliptic Curve Cryptography

The encryption system which is used in this project is the ElGamal, which is built on the Diffie-Hellman Key exchange. This can be defined over any cyclic group. The level of

security depends on a certain problem relating to discrete logarithms.
The steps used in this encryption process have been explained above, like the Diffie-Hellman key exchange algorithm, modular arithmetic, Congruence and Inverse Modulo.
The only difference between this algorithm and the Diffie-Hellman key exchange algorithm is the coordinates and computations in the elliptic curve.
The encryption and decryption process can be grouped into three main parts, which are:

5.2.1 Generating Public and Private Keys

The equation for the elliptic curve is $y^2 = x^3 + ax + b$, where a and b are constants that determine the type of equation, some curves are vulnerable to attacks whereas others are highly secured, that is the constants a, b can determine how secured the elliptic curve is, the security of elliptic curve cryptography depends on the difficulty in solving the elliptic curve logarithm problem.

Both parties A and B agree on the limit or range to work in, as well as the constants a, b these parameters determine an elliptic group E (a, b) to work in.

With the agreed parameters and limit set, the elliptic group of points can be determined or generated. The algorithm used to generate the points is explained above.

The parties can then select a point G, in the elliptic group, this point can be made public to both parties and even a third person.

With the point G selected, both parties can now generate a public key for the other party. This is done by selecting a private key $n_{x,}$ by both parties such that n_x is only known to the party that selected it. The public key is then generated by multiplying the private key with the point G agreed by the two parties.

For example, if party A has to generate a public key P_A for the other party we have $P_A = n_A * G$.

Both parties can then announce their public keys to the other parties for encryption and decryption of data, but the private key is kept secret

5.2.2 Encryption

To encrypt and send any message, P_m from one party to the other, the sender takes the public key of the receiver and multiplies it with his own private key $n_{x,}$ to generate a shared secret between the two parties. This shared secret is not public and only known to the two parties involved.

The shared secret is added to the message to get an encrypted message P_E, which can be sent to the other parties.

In sending the encrypted message, a cipher text $C_m = \{Px_{,} P_S\}$ is sent to the receiver for decryption.

For example if party B has to encrypt a message to person A, he first multiplies his private key n_B with the public key of person A, P_A from the example above to generate a shared secret, P_S. The message P_m is then added to the shared secret P_S to get the encrypted message $P_{E,}$ such that $P_{E =} P_m + n_{B*} P_A$. Party B then send the cipher text $C_B = \{P_{B,} P_S\}$ to party A for decryption.

5.2.3 Decryption

To decrypt the message, the receiving party, takes the public key of the sending party Px and multiplies it with his private key n_x the result is the shared secret which is known to only the two parties A and B. With the secret key the receiver can then subtract it from the encrypted message, to get back the original message Pm.

For example if party A has to decrypt a message from person B, he first multiplies his private key n_A with the public key of person B, P_B to generate a shared secret, P_S. The shared secret P_s is then subtracted from encrypted message $P_{E,}$ to get back the original messagesuch that $P_{m =} P_E - n_{A*} P_B$.

It is very difficult for a third part to decrypt the message given the cipher text alone. For instance, for a third-party C to decrypt the message given the cipher text he needs to know the private key n_x of the receiving party to generate the shared secret. This means given the public key of the receiving party and the point G, the third party must find the multiplier that creates public key. The method used for this known as the elliptic curve logarithm problem and the only method available to solve it is the Polard rho algorithm, which is infeasible if the private key or the elliptic group GF (p) is large.

5.3 Materials and Tools

The Dev C++ IDE was used for the development of the application. C++ was the main programming language used. The application can run on a network of computers with a minimum processor type Pentium IV or better, 10GB Hard drive capacity or better and 1GB RAM or better

6 OBSERVATION

6.1 Key Length Comparison of RSA and ECC

The length of the key used in encryption has a direct relationship with the computer resources needed and how fast the encryption or decryption would be. From Table 3 and Figure 4.9, it is observed that ECC key length size is very small as compared to RSA providing the same level of security. This shows that using ECC to encrypt data would be faster as compared to RSA.

Table 3 Key Length Comparison of RSA and ECC

Security Level(bits)	**RSA key length (bits)**	**ECC key length (bits)**	**Approximate ratio**
80	1024	160-223	5-6:1
112	2048	224-255	8-9:1
128	3072	256-283	11-12:1
192	7680	384-511	15-20:1
256	15360	512-571	27-30:1

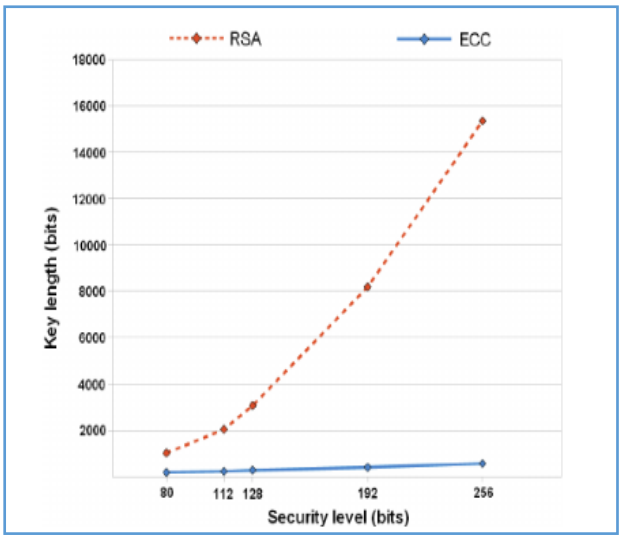

Fig 5. Key Length Comparison for RSA and ECC Cryptosystems

6.2 Performance Evaluation of RSA and ECC

The ECC performs better as show in Fig. 6 due to its key length being relatively smaller and also ensures much security due to the algorithm used for the encryption.

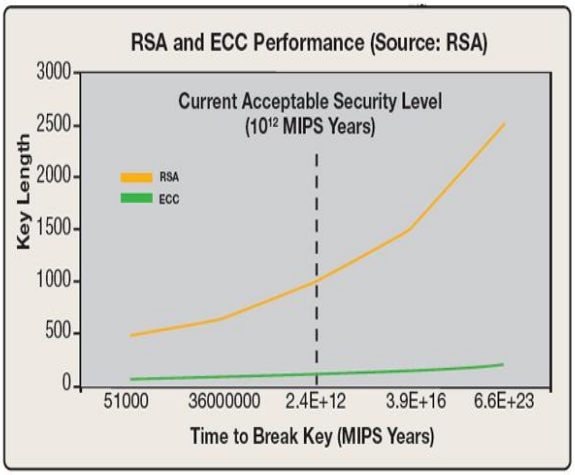

Fig 6. Performance of RSA and ECC

6.3 ECC and RSA Comparison

The various parameters used in verifying the performance are illustrated in the Tables 4, 5, 6 and 7. The performance indicators used include Key Generation, Signature Generation and Signature Verification.

Table 4. Key Generation Performance

Key Length(bits)		**Time(s)**	
RSA	ECC	RSA	ECC
1024	163	0.16	0.08
2240	233	7.74	0.18
3072	283	9.80	0.27
7680	409	133.90	0.64
15360	571	679.06	1.44

Table 5. Signature Generation Performance

Key Length(bits)		**Time(s)**	
RSA	ECC	RSA	ECC
1024	163	0.01	0.15
2240	233	0.15	0.34
3072	283	0.21	0.59
7680	409	1.53	1.18
15360	571	9.20	3.07

Table 6. Signature Verification Performance

Key Length(bits)		**Time(s)**	
RSA	ECC	RSA	ECC
1024	163	0.01	0.23
2240	233	0.01	0.51
3072	283	0.01	0.86
7680	409	0.01	1.80
15360	571	0.03	4.53

Table 7. ECC and RSA Overview

Parameters	**ECC**	**RSA**
Computational Overheads	Roughly 10 times that of RSA can be saved	More than ECC
Key Sizes	System parameters and key are shorter for the ECC	System parameters and key pairs are larger for the RSA
Bandwidth saving	ECC offers considerable bandwidth savings over RSA	Much less bandwidth saving than ECC
Key Generation	Faster	Slower
Encryption	Much faster than RSA	At good speed but slower than ECC
Decryption	Much faster than RSA	Faster than ECC
Smell Devices efficiency	Much more efficiency	Less efficient than ECC

7. CONCLUSION

The results provided during the analysis of Elliptic Curve Cryptography (ECC) as against the Rivest-Shamir-Adleman (RSA) and the other types of encryption algorithms, it is seen that, ECC is much more efficient than the other encryption algorithms. Elliptic Curve Cryptography has a shorter key length and size which is an important security parameter. It also saves bandwidth which facilitates key generation in the

encryption and decryption of data hence enhancing performance. Moreover, ECC ensure faster encryption and decryption, and efficient even in small devices. From all the explicitly stated results, it can be stated without an iota of doubt that Elliptic Curve Cryptography is a better option to use in securing data.

8. ACKNOWLEDGMENTS

Our thanks to John Amfo and other experts who have contributed towards this paper.

9. REFERENCES

[1] Stallings, W. (2005). Cryptography and Network Security Principles and Practices. Prentice Hall.

[2] Singh, G., & Supriya. (April 2013). A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security. International Journal of Computer Applications, 67(19), 0975 – 8887.

[3] Fielding, R. (1999). Hypertext Transfer Protocol – HTTP/1.1. RFC 2616.

[4] Frier, A., Karlton, P., & Kocher, P. (n.d.). The SSL3.0 Protocol Version 3.0. Retrieved from http://home.netscape.com

[5] Wagner, D., & Schneier, B. (1996). Analysis of the SSL 3.0 protocol. 2nd USENIX Workshop on Electronic.

[6] Ylonen, T., Kivinen, T., Saarinen, M., Rinne, T., & Lehtinen, S. (2003). SSH Protocol Architecture..