Electron Commer Res (2010) 10: 27–41
DOI 10.1007/s10660-010-9047-z

Improving e-payment security using Elliptic Curve Cryptosystem

O.R. Vincent · O. Folorunso · A.D. Akinde

Published online: 24 March 2010

Abstract The use of e-commerce has been associated with a lot of skepticism and apprehension due to some crimes associated with e-commerce and specifically to payment systems. The secure socket layer (SSL) protocol is trusted in this regard to secure transactions for sensitive applications like e-commerce. Unfortunately, the use of SSL protocol causes slow response time on the server which is a major cause of frustration for on-line shoppers. In this paper, we propose a secured credit-debit card payment systems based on Elliptic Curve Cryptosystem (ECC). We first examined ECC algorithm over prime fields $GF(p)$, implement our proposed method using a typical transaction involving credit/debit card numbers and compared the performance with RSA cryptosystem. Our result shows that ECC is faster in terms of response to transaction request and occupies less memory space than equivalent RSA system. Thus, these makes it more suitable public Key cryptography scheme for application in a constraint open environment like payment system where fast operations are needed.

Keywords Elliptic Curve Cryptography · E-commerce · E-payment · Security · Prime finite field $GF(p)$

O.R. Vincent (✉) · O. Folorunso · A.D. Akinde
Artificial Intelligent Group, Department of Computer Science, University of Agriculture, P.M.B. 2240, Abeokuta, Nigeria
e-mail: vincent.rebecca@gmail.com

O. Folorunso
e-mail: folorunsolusegun@yahoo.com

A.D. Akinde
e-mail: adakinde@lagosmainlanddiocese.org

Present address:
O.R. Vincent
Computational Intelligent Group, Institute of Informatics, Clausthal University of Technology, 38678 Clausthal-Zellerfeld, Germany
e-mail: rebecca.vincent@tu-clausthal.de

1 Introduction

The notion of the world as a global village is not a new one as the developments in computer and communication technologies has not only afforded us with global information and resource sharing capabilities; but also the ability to do business from remote locations. It is unfortunate that this new frontier in business known as e-commerce has unequivocally suffered from attack from various sources. These attacks, usually directed at e-commerce user's payment information wherever they might be located, either on the client side or the server side (in data repositories) or in locations between this two (network, or facilitator's hardware) have had its toll of enervating effects on users and site owners alike.

E-commerce provides the capability of buying and selling products, information and services on the Internet and other online environments. Just like any trading activity, the safety and reliability of money exchange between transacting parties is essential. Electronic payments being an integral part of e-commerce are the most critical aspects when it comes to safety of money, security of information, etc. Generally defined, electronic payment is a form of financial exchange that takes place between the buyer and seller facilitated by means of electronic communications in an online environment [11]. Electronic payment systems (EPS) are summoned to facilitate the most important action after the customer's decision to pay for a product or service—to deliver payments from customers to vendors in a most effective, efficient and problem-free way.

The need for online payments was first addressed by using extant payment methods of the offline world for online payments. For example credit cards, originally intended as an offline credit instrument, have become the major payment instrument for e-commerce. As e-commerce and online purchasing grows, the weaknesses of credit-debit cards and cheques are becoming more apparent. In the credit-debit cards approach, information representing money is usually transferred electronically between parties over computer networks.

One of the most crucial and well-researched issues in payment systems is security. This is because the Internet being an open network with no centralized control is such that, the infrastructure, supporting e-commerce and payment systems in particular, must be resistant to attacks in the Internet environment [4]. E-commerce plays an important role in the word's global economy. The dominant security protocol for handling e-commerce over the internet has been Secure Socket Layer (SSL) protocol which has its technology based on public key cryptography (PKC) [14, 29]. In PKC, Rivest, Shamir and Adleman (RSA), Digital Signature authentication (DSA) and Diffie-Hellman [7], protocols are used traditionally. Unfortunately, the use of these schemes imposes significant performance penalty on web servers and has a great impact of web performance on e-commerce transaction while small devices like cell phones are beyond the scope.

To this end, there is a need for more and smaller devices with transaction volume; need for more privacy and therefore, the demand for increased security. Security in e-commerce can be viewed as a two-fold issue. On the one hand, users would like to be sure that their money is safe when paying online. On the other hand, banks and payment service organizations would like to protect themselves so that no money, financial, or personal information can be stolen or misused.

It is therefore not strange that despite the ubiquitous use of e-commerce technologies, its level of acceptance has not been at its peak. This is most likely due to skepticism arising from security issues. Various methods have been applied to address the situation and there have also been considerable amount of research on this crucial area. However, it is quite evident that the success rate has not commemorated with security task involved with e-payment crimes. Cryptography has been a major tool for data security in e-commerce sites and its effectiveness, efficiency, ease of use; inter-operability etc. is of great interest in this work. This paper is aimed at providing an alternative and yet more efficient approach to addressing the problems of security in e-payment using elliptic curve cryptography. Using our method, we give supporting evidence for the efficiency of ECC for achieving improved and secured e-payment involving credit-debit cards.

2 The need for ECC

When exchanging financial, business or personal information, clients want to ensure that the information is neither modified (data integrity) nor disclosed (confidentiality) in transit. The popular protocol used for this purpose is the Secure Socket Layer (SSL) supported by Visa and Master card. The reason why the deployment of SSL protocol outpaced other security protocols such as SSH [30], S/MIME [22] and SET [16] is that the SSL protocol is application independent. Though, there are many examples of application protocols like TELNET, FTP, IMAP and LDAP running transparently over SSL, the most common usage of SSL is for securing HTTP [8], the main protocol of the World Wide Web. SSL is trusted to secure transactions for sensitive applications ranging from web banking and stock trading, to e-commerce.

Unfortunately, the use of SSL protocol causes slow response time on the web server [29]. In [6], it was reported that secure web servers run between 3.4 to 9 times slower compared to regular web servers on the same hardware platform. A major cause of frustration for on-line shoppers is slow response time and this often leads to abandoning the electronic shopping carts during check out. According to a research report in [31], the potential revenue loss from aborted e-commerce transactions due to web performance issues exceeds several billion dollars annually.

In its most common usage, SSL utilizes Rivest, Shamir and Adleman (RSA) public key scheme encryption to transmit randomly chosen secret that is used to derive keys for data encryption and authentication [23]. The RSA decryption operation is the most computationally intensive part of an SSL transaction for a secure web server [1]. Although, RSA is highly secure and widely used, there are some potential problems with the use. The only way to attack it is to perform a brute-force attack on the modulus. This attack can be easily defeated by simply increasing the size. However, this approach can lead to a number of problems, among which are increased processing time and increase key storage requirement since RSA key storage require significant amount of memory for storage. In addition, key generation is complex and time consuming.

The above issues could constitute major problems especially for devices with limited memory capacity and processing power, such as smart cards or mobile phones.

Consequently, an alternative public key algorithm with the same high level of security, with relatively short keys is desirable. Motivated by these, we present in this paper, an alternative algorithm based on mathematical objects known as elliptic curves, which can in some circumstances provide significant benefit over the application of RSA for credit-debit card transaction security.

3 Background to ECC

Elliptic curve cryptography was proposed by Koblitz [12] and Miller in 1985 [19]. They implemented existing public key algorithms like Diffie-Hellman using elliptic curve. Since then, a lot of scholars have researched into this area from different perspectives and for different applications. For instance, Chen et al. [5] designed and implemented fast algorithms using elliptic curve cryptography over the field $GF(p)$ in relation with modular addition, subtraction, point addition, point production and choice of embedding plaintext to a point.

Recently, Hedabou et al. [9] proposed a security scheme to avoid side channel attacks using ECC by combining an optimized use of space memory with high level of security and efficiency. Shi and Yan [24] studied and identified the problems in the software implementation of ECC and explored techniques that can accelerate the software implementation, however, their work was based on ECC in $GF(2^m)$. In addition, Vijayalakshmi and Palanivelu [27] used elliptic curve cryptography to investigate secure localization in wireless sensor networks. The challenges imposed on wireless sensor networks with regards to communication between authenticated neighbours and their precise locations in a secure manner were identified; and furthermore, the problem of insecurity in sensor networks was addressed. Vijayalakshmi and Palanivelu [27] also showed that ECC satisfies all the constraints of the sensor networks- minimum bandwidth, power, energy and computational speed; and thus suitable for secure localization in sensor networks.

Jena et al. [10], proposed a novel protocol for smart card using ECDLP (Elliptic Curve Discrete Logarithm Problem) for securing entity authentication, data integrity and confidentiality. The secure channel protocol described by Jena et al. [10] uses a combination of secure public key system and secret key to achieve the desired output. This proposed scheme however, was not implemented for payment system. Another related work to this paper, is the one of Vivek et al. [28] which described ECC and compared its strength with that of RSA. In Vivek et al. [28], a possible scheme for RSA encryption was presented and its performance was compared with that of ECC. Their results showed that ECC is faster; occupies less memory than an equivalent RSA system; and would be most suitable for smart cards. In addition, their studies clearly suggested that EC cryptographic would need to be further investigated in the context of smart cards so as to ascertain its suitability for wide varieties of systems. Thus, the current paper is a step-forward in the direction of a new elliptic curve cryptosystem using ECDLP algorithm that is aimed at securing information on credit-debit cards.

3.1 Overview of Elliptic Curve Cryptosystem

Elliptic curves used in cryptography are typically defined over two types of finite fields: the fields of odd characteristics $GF(p)$, where $p > 3$ is a large prime number and the fields of characteristics two $GF(2^m)$ called binary fields [3]. We define our elliptic curve over the field of odd characteristics. If the set of all pairs of affine coordinates (x, y) for $x, y \in F_q$ form the affine plane $F_q \times F_q$ [3, 20], the locus of points in the affine plane whose coordinates satisfy

$$y^2 = x^3 + ax + b \tag{1}$$

is a case of an elliptic curve with a point at infinity O, where characteristic $p > 3$ and $a, b \in F_p$ are constants such that $4a^3 + 27b^3 \neq 0$.

Let $K = F_q$ be a finite field and $f(x) \in K(x)$ be a cubic polynomial with coefficients in K which has distinct root and $q > 3$ is a prime integer. Then, the solutions to (1) is

$$y^2 = f(x) \tag{2}$$

where x and y are in some extension K^1 of K, are called the K^1-points of the elliptic curve defined by (2) [13]. According to Hasse's theorem on elliptic curve, the number of points on a curve is close to the size of the underlying field such that $(\sqrt{q} - 1)^2 \leq |E(F_q)| \leq (\sqrt{q} + 1)^2$ [25]. The points on an elliptic curve form an abelian group $(E(F), +)$ with O called the infinity point. With O as a group identity under addition, the set of points E form a finite commutative group [21, 26]. We follow the rational formula for the addition rule involving many arithmetic operations as given in [5] and [15]. If we define the negative of $P = (x_1, y_1) \in GF(p)$ to be $-P = (x_1, -y_1)$ and $Q = (x_2, y_2)$ with $Q \neq -P$, then $(P + Q) = (x_3, -y_3)$ can be calculated as

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2, \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned} \tag{3}$$

and

$$\begin{aligned} \lambda &= \frac{y_2 - y_1}{x_2 - x_1}, \quad x_1 \neq x_2 \text{ (adding)}, \\ \lambda &= \frac{3x_1^2 + A}{2y_1}, \quad x_1 = x_2 \text{ (doubling)} \end{aligned} \tag{4}$$

Since the preceding formulas for adding and doubling of elliptic curve points need a field inversion, which is more expensive than the field multiplication [9, 18], we represent the points using projective coordinates to avoid the need for an inversion.

3.2 Encrypting credit-debit cards message with ECC

In this scheme, a problem based on elliptic curve discrete logarithms problem (ECDLP) is considered. Given an elliptic curve E defined over a finite field $GF(p)$, a point $P \in E$ of order n, and a point $Q = kP$ where $O \leq k \leq n - 1$, here, k is to be determined. In a prime case of ECC, the domain parameters are (p, a, b, G, n, h)

where n is the size of a subgroup of $E(F_q)$, it follows from Lagrange's theorem that the number $h = \frac{|E|}{n}$ is the integer called the cofactor, where $h \leq 4$ [2].

Algorithms 1 and 2 are steps for implementing ECC for electronic payment systems. While Algorithm 1 is an integer multiplication, in the Algorithm 2, a plaintext m is embedded to a point of the elliptic curve E before transmitting the message m. If m is an integer such that $0 < m < p/1000 - 1$, three digits are appended to m to obtain integer x of such when $1000m \leq x < p$ with (1) becoming $f(x) = x^3 + ax + b = x_m \bmod p$ being a quadratic residue. Then by obtaining the solution for $y^2 = f(x) \bmod p$, we get the plaintext message point $P_m = (x, y)$ on E.

Preposition 1 *If mult*: $Z^+ \times Z^+ \to Z^+$ *is defined by* $mult(a, b) = ab$ *then* $mult \in FP$.

Proof For a binary integer a, we simply shift all of its digits left by a single place and add a zero to the right of a. We denote this operation by $2 \times a$, then consider the following algorithm for integer multiplication [17]. □

In the algorithm, the loop denoted by (2) above is repeated at most n times and each line of the algorithm involves basic polynomial time operations on integers with at most $2n$ bits, hence $mult \in FP$.

We continue in this way until either we find a square root or reach $j = k$. If we reach $j = k$ and y does not have a square then we fail to embed our message as a point

Algorithm 1 Integer Multiplication

Input: n-bit binary integers $a = a_n \ldots a_1$ and $b = b_n \ldots b_1$.
Output: $mult(a, b)$ in binary.
(1) $m \leftarrow 0$
(2) for $i = 1$ *to* n
(3) if $b_i = 1$ then $m \leftarrow m + a$
(4) $a \leftarrow 2 \times a$
(5) next i
(6) output m.

Algorithm 2 Algorithm for ECC based E-payment

Input: Plaintext m
Output: P_m
(1) Convert plaintext to be encrypted to integer m where $m \to P_m$
(2) Let k be an integer between 30 and 50
(3) Suppose message m is an integer with $(m + 1)k < q$
(4) Integer $mk + j \in F_q$ where $0 \leq j < k$ for $j = 1, 2, \ldots k$
(5) Obtain an element $x \in F_q$ that correspond to $mk + j$
(6) If q is a prime, then compute $y^2 = x^3 + ax + b$
(7) Calculate $y = \sqrt{x^3 + ax + b}$
(8) If there exist square root of y, then we take $P_m = (x, y)$
(9) Otherwise, $j = j + 1$
(10) Continue.

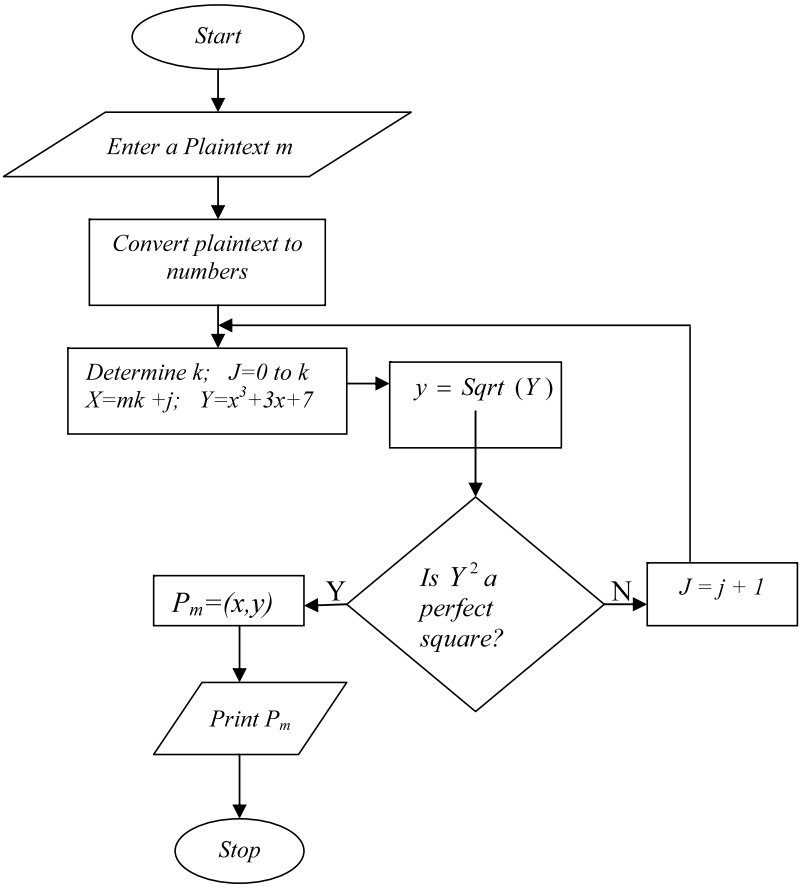

Fig. 1 Process flow through the EC Cryptosystem

on the elliptic curve and the probability of failure is about $\frac{1}{2^k}$. In order to recover the message from the point $P_m = (x, y)$ we calculate m, defined by $m = \lfloor \frac{x}{k} \rfloor$, where $\lfloor \frac{x}{k} \rfloor$ is the greatest integer less than or equal to $\frac{x}{k}$. The process flow for encrypting the message is given in Fig. 1.

4 Implementation

The elliptic curve based cryptosystem for e-payment described in Sect. 3 was implemented using Java Creator with JDK 1.6. For our prime field $GF(p)$, p was chosen to be Mersenne prime with bits size being a multiple of 32—a Mersenne prime of $2^n - 1$, where n is a prime. For each cipher suite, we studied three different security levels—160, 192, and 224 bits for ECC and 1024, 1536 and 2048 bits for RSA. For the ECC we chose three elliptic curves recommended by NIST and Federal Information Processing Standard (FIPS). The curves are secp160r1, secp192r1 and secp224r1 defined over prime integer fields.

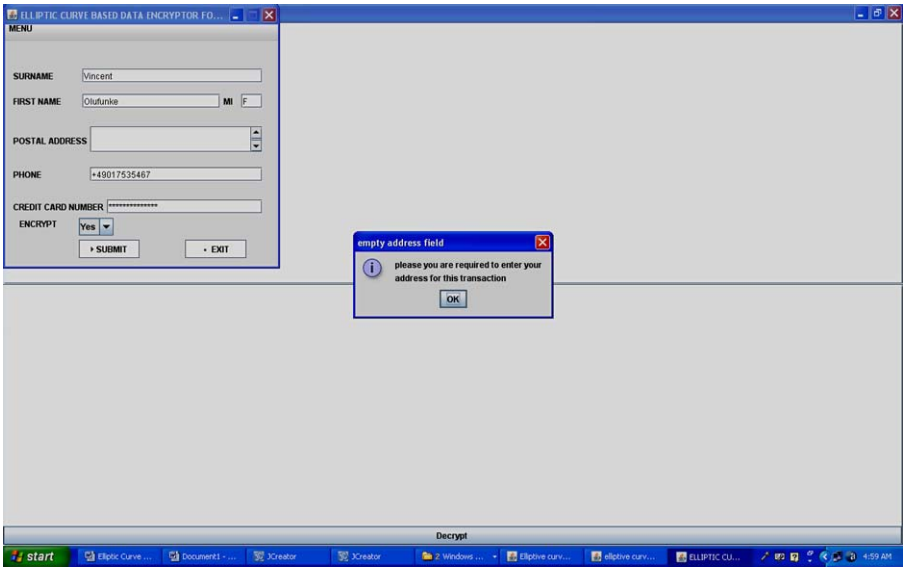

Fig. 2 Transaction interaction between the client and the server

The ECC operates over a group of points on an elliptic curve. The protocol relies on solving the elliptic curve discrete logarithm problem (ECDLP) which state that given P and Q such that $Q = kP$, find k. This means a point P multiplied by an integer k resulting in another point Q on the curve. The scalar multiplication is performed through a combination of point-additions and point-doublings. The important elliptic curve parameter G (base point) which is fixed for each curve is multiplied with a large random integer k (private key) to result in the corresponding public key.

Considering the credit-debit card situation where the card numbers are made up of Batch Identification Number (BIN) and Personal Identification Number (PIN) in a transaction environment, we have a number of n users making transactions; the users can interact with the system by supplying some personal information like name, address and the credit or debit card numbers. The fourteen numbers on the credit card (BIN and PIN) are encrypted for security during the transaction process. Figure 2 shows how n users can interact with the ECC based transaction system to purchase goods and services online using cards. In between the two major players: the cardholder called the client and administrator, there exist a major abstraction of processes through which the integrity of data transferred can be compromised if not encrypted. In Fig. 2, the user is given a choice to send either plain or encrypted credit card numbers.

The user could send encrypted numbers while the administrator on the other hand residing at the server facilitates the transaction and will be ready to cope with both encrypted and plaintext credit card numbers. The transaction manager is the only one with the key in this situation and can decrypt the encrypted message at the server as shown in Fig. 3, where a user sends an encrypted message and the representation of the message is shown at the backend of the server.

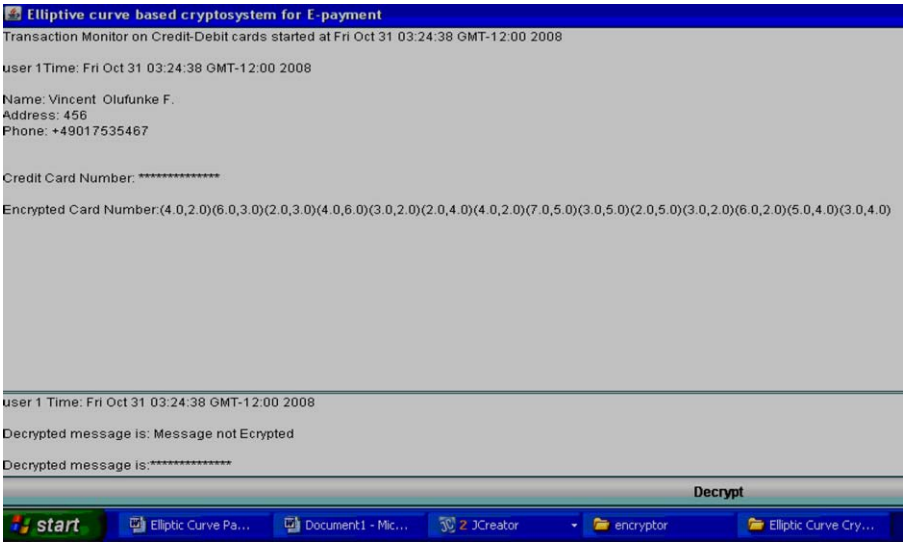

Fig. 3 Encrypted and decrypted message on the server

5 Evaluation

We evaluated the performance of our proposed ECC for e-payment and compared it with RSA in terms of key size, computational power, operation speedup, signature generation and verification; and encryption and decryption. These experiments were carried out on an Intel Centrino Core 2 Duo with 1.8 GHz, and 512 MB RAM running on Linux OS. In our implementation, Java and Bouncy Castle Java API's version 1.23 with the class comparison .java were used.

5.1 Computational power

To give a comparative performance analysis in terms of computational power of our model, we implemented both RSA and the ECC algorithms for credit-debit cards transaction using the PIN numbers and observed the response times of our ECC for every transaction request and that of RSA. The prime fields with p being a Mersenne prime allows for efficient modulation reduction. It was observed that ECC-160 provides the same security as RSA-1024. Similarly, ECC-192 and ECC-224 provide same security with RSA-1536 and RSA-2048, respectively as recommended by NIST. The performance of RSA and ECC for each security level is given in Table 1. The result shows that ECC produces more efficient computation speed compared to RSA. Hence, its design in securing card payment system proofs to be superior scheme in terms of its efficiency. Considering ECC-160 and RSA-1024, we plot a graph of response time against request per second for every transaction requested in Fig. 4.

Figure 4 shows that the use of ECC allows the server to handle a large number of requests compared to RSA. In addition, ECC responds faster to request than RSA. These advantages are particularly important in constraint environment such as the e-payment transactions. In Fig. 4, we observed that the saturation point of the server

Table 1 Performance of RSA and ECC

	RSA-1024		ECC-160	RSA-1536		ECC-192	RSA-2048		ECC-224
Time (ms)	9.79		3.67	22.24		2.63	61.62		6.08
Performance ratio	1	:	2.6	1	:	8.5	1	:	10.1
Key-size ratio	6.4	:	1	8	:	1	9.14	:	1
Speedup	1	:	17.5	1	:	30.1	1	:	48.7

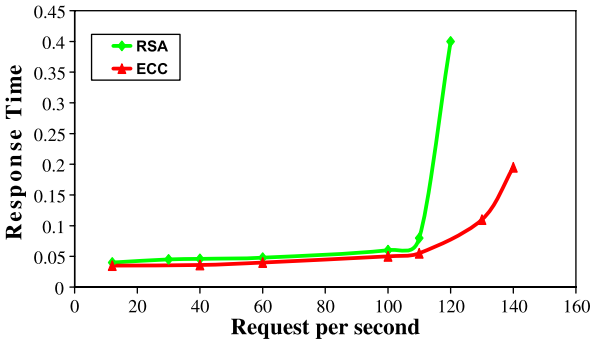

Fig. 4 Response time v/s transaction request for ECC and RSA

is reached earlier with RSA which resulted in sharp increase in latency at 107 requests. The slow latency exhibited by ECC is less than 48% of the RSA case. We also observed that between 10–100 transaction requests, both RSA and ECC behave in similar manner with negligible difference in response time. This implies that for small transactions, the performance of RSA is comparable to that of ECC; whereas for larger transaction requests (typically $>$ 100), RSA reaches its optimal (saturation point) earlier than ECC. This obviously gives further justification for the higher security strength exhibited by ECC.

5.2 Signature generation and verification

It is possible to define an elliptic curve analog to RSA cryptosystem that are based on discrete logarithm problem such as Digital Signature Algorithm (DSA). This offers practically essential advantage over RSA. In our experiment, we consider an analog to DSA described as elliptic curve DSA (ECDSA). The mechanisms for signature generation and verification were devised to depend on elliptic curve discrete logarithms and ECC library was created for ECDSA operations. According to NIST and FIPS 182-2 recommendation, ECDSA over prime fields is used as the benchmark in the experiment. The algorithm for the implementation was designed to permit users to change to different curves, and in particular key sizes. This is aimed at making the system flexible to fit into restricted environments like payment systems. For each implementation, computation times were recorded. Table 2 shows the results obtained for three different keys with equivalent security. The results from Table 2 clearly show that ECDSA is faster in generating signature than RSA.

Table 2 Signature generation and verification of ECC and RSA

	Signature generation (ms)	Signature verification (ms)
ECDSA-$F_{p=160}$	41.14	39.81
RSA-1024	81.22	5.05
ECDSA-$F_{p=192}$	45.59	42.92
RSA-1536	504.06	15.19
ECDSA-$F_{p=224}$	48.76	43.12
RSA-2048	1254.42	78.60

Table 3 Encryption and decryption of ECDSA compared with RSA

	Encryption (ms)	Decryption (ms)	Total (ms)
ECDSA-$F_{p=160}$	39	38	77
RSA-1024	1	22	23
ECDSA-$F_{p=192}$	52	51	103
RSA- 1536	2	69	71
ECDSA-$F_{p=224}$	66	63	129
RSA-2048	3	130	135
ECDSA-$F_{p=256}$	48	49	97
RSA-3072	5	308	313
ECDSA-$F_{p=384}$	112	124	236
RSA-7680	36	4956	4992
ECDSA-$F_{p=512}$	522	587	1109
RSA-15360	162	38928	39090

5.3 Encryption and decryption

To compare the encryption and decryption, we used six different key sizes that are equal in terms of security to show possible process shift between the systems. We measure the speedup in the execution time of ECC and RSA encryption and decryption. The number of RSA encryption that a server can perform within a certain time period was determined. We observed that the most computationally expensive operation of the RSA protocol is the server's private key decryption due to modular multiplication. Table 3 shows the results of the various keys used to test both performances with different security parameters.

In order to ensure reliable results, encryption and decryption were executed 20 times and the average performance times were recorded. The data of Table 3 is plotted in Fig. 5. In the plot shown in Fig. 5, a time up to 700 ms was chosen considering RSA decryption with large values and encryption with very low values. Notice that in the chosen range, we could not include two other extreme and large values—these could be read-off from the data in Table 3. By observing also the trend lines for each plot of Fig. 5, we discovered that each curve follows a polynomial function of order $n \geq 4$. These are not shown for brevity. In addition, we noticed that the keys grow exponentially with computation time. Again, while the trend line for ECC decryption

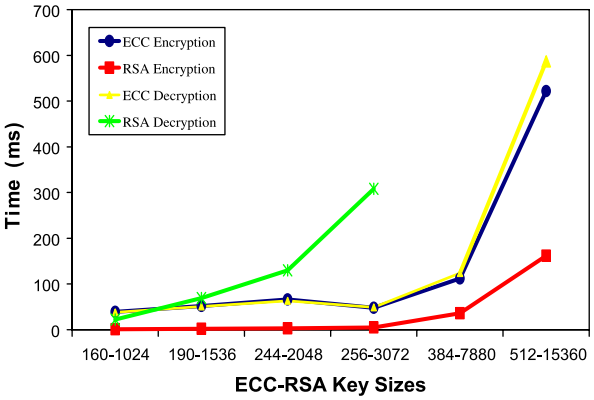

Fig. 5 Encryption/decryption comparison of ECC and RSA

is 4th-order polynomial, the trend line for RSA decryption is polynomial of order 5. The implication of the difference in the polynomial order for decryption curves is an issue that could not be ascertain in our study. We believe, however, that this could be related to the difference in the saturation points discussed earlier. Though, the plot shows that encryption process in RSA is optimal even for large key sizes such as 7680 bits and 15360 bits, the decryption indicates that the complexity of RSA rises exponentially. Thus, for key sizes greater than or equal to 2048 bits it is reasonable to conclude that ECC outperforms RSA.

6 Discussion

A major advantage of ECC over RSA is that the basic operation in ECC is point addition, which is known to be computationally very expensive. This constitutes one reason why it is very unlikely that a general sub-exponential attack on ECC will be discovered in the near future, though ECC has a few attacks on a few particular classes of curves. These curves can be readily distinguished and can be avoided. Most attacks on ECC are based on attacks on similar discrete logarithm problems, but these work out to be much slower due to the added complexity point addition.

On the other hand, RSA exponentiation involves a sequence of modular multiplications with already known sub-exponential attack which works in general. Thus, to maintain the same degree of security, with respect to rising computing power, the number of bits required in the RSA generated key pair will rise much faster than in the ECC generated key pair (see Table 1). The key generation for ECC outperforms RSA at all key lengths, and is more pronounced as the key length increases. One of the reasons why ECC generates private/public key pairs in superior speed to RSA is that it does not have to devote resources to the computationally intensive generation of prime numbers. Hence, ECC key generation time grows linearly with key size, while key generation time for RSA grows exponentially.

Encrypted message is a function of key size and data size for both RSA and ECC. Since ECC key size is relatively smaller for RSA key size, encrypted message in ECC is smaller. As a result, computational power is smaller for ECC. The faster compu-

tation of ECC alleviates the computational burden on secure web servers. A characteristic measure for the performance of a secure e-payment system is the rate at which it can service its transaction connections. As observed in Fig. 4, performance on ECC in terms of lower average time taken by the server to complete a transaction request for different transaction requests when compared to RSA give further proofs of its efficiency for electronic transactions. After estimating the relative costs with the overall processing time, RSA decryption continues to be the dominant cost in all of these cases (see Table 2). Hence, the model incorporated ECC in e-payment because transporting sensitive data like credit/credit card numbers and personal information of the customers or cardholders over the internet where high speed requires strict security measures. By replacing the traditional public-key cryptosystem with ECC, the speed of web transactions, including the payment systems would increase; and thus would also enhance web server.

In any smart card like credit card, more transistors are required to perform computation especially for higher bit encryption which increases the processor used. ECC could reduce the number of transistors because the number of processors involved in ECC is much smaller than in RSA system. Thus, this serves as an added advantage for using ECC to secure credit/debit payment cards transactions.

7 Conclusion

In this paper, a practical implementation of elliptic curve cryptosystem over the field $GF(p)$ in electronic payment system using credit/debit cards payment is presented. Our protocol is based on e-payment application where we consider real life electronic card payment for our implementation and encrypt a typical credit card numbers. In electronic transaction, ECC-based payment system is essential for many purposes such as protecting the transactions against attack on the network, ensuring the security without prior arrangements between customers and vendors, guaranteeing the transaction integrity; and authenticating the clients and the banks.

The keys generated by the implementation are highly secured and consumes lesser bandwidth as a result of smaller size used by elliptic curves. The advantage of ECC over existing cryptosystem is that while it provides the same security with other public key cryptosystem like RSA, ECC provides higher computational speed which is needed in e-payment system. Hence, ECC offers equivalent security in e-commerce payment system with much smaller key size, faster response time, low power consumption, low memory usage and low CPU utilization.

Furthermore, a detailed examination of the signature generation and verification as well as encryption and decryption times for both the ECC using the ECDSA algorithm and RSA also show that ECC algorithm is better for e-payment systems and further provides a supporting evidence of its efficiency over RSA algorithm. Based on our results, we conclude that ECC would be most suitable public key cryptography scheme for a constraint and open environment like payment systems. Its efficiency, effectiveness and ease of use in security makes it an attractive scheme and could therefore serve as an alternative to other cryptography. The above properties could enhance e-commerce security on the Internet and hence, make electronic transaction and other e-business to be carried out with little or no fear of hackers.

Acknowledgements We acknowledge the valuable comments of the reviewers and also appreciate useful discussions with Peter Novak of the Institute of Informatics, Clausthal University of Technology.

References

1. Adewumi, A. O., Longe, H. O. D., Uwadia, C. O., & James, T. Y. (2000). Security issues in electronics: a close look at RSA and elliptic curve cryptographic algorithms. In *Proceedings of the 16th national conference of the computer association of Nigeria (COAN) on deployment of telematics systems: trends, techniques and tools (Telematics 2000). COAN Conference Series* (Vol. 11, pp. 187–202).
2. Akinwande, M. B. O., Idowu, B. A., Olaitan, H. M., & Ogungbe, A. S. (2006). Digital comparison of elliptic curve and RSA public-key cryptosystems. *Journal of Computer Science and its Applications*, *12*(1), 91–99.
3. Blake, I. F., Seroussi, G., & Smart, N. P. (1999). *Elliptic curves in cryptography*. New York: Cambridge University Press.
4. Chaum, D. (1992). Achieving electronic privacy. *Scientific American*, *4*, 96–101.
5. Chen, L., Yanpu, C., & Zhengzhong, B. (2004). An implementation of fast algorithm for Elliptic Curve Cryptosystem over $GF(p)$. *Journal of Electronics*, *21*(4), 346–352.
6. Coarfa, C., Druschel, P., & Wallach, D. (2006). Performance analysis of TLS web servers. *ACM Transactions on Computer Systems (TOCS)*, *24*(1), 39–69.
7. Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, *22*, 644–654.
8. Fielding, R., et al. (1999). Hypertext transfer protocol—HTTP/1.1. *RFC 2616*.
9. Hedabou, M., Beneteaus, L., & Pinel, P. (2008). Some ways to secure Elliptic Curve Cryptosystem. *Advances in Applied Clifford Algorithm*, *18*, 677–688.
10. Jena, J., Pornography, S. K., Biswal, P. K., & Jena, S. K. (2008). A novel protocol for smart card using ECDLP. In *1st International conference on emerging trends in engineering and technology* (pp. 838–843). Los Alamitos: IEEE Computer Society.
11. Jonathan, B., David, K., & Daniela, R. (1998). Market-based resource control for mobile agents. *Proceedings of Autonomous Agents*, 197–204.
12. Koblitz, N. (1987). Elliptic Curve Cryptosystems. *Mathematics of Computations*, *48*, 203–209.
13. Koblitz, N. (1993). *Introduction to elliptic curves and modular forms*. Berlin, Heidelberg, New York: Springer, ISBN 3-540-97966-2
14. Li, S., Wu, Y., Zhou, J., & Chen, K. (2008). A practical SSL server performance improvement algorithm based on batch RSA decryption. *Journal of Shanghai Jiaotong University (Science)*, *13*(1), 67–70.
15. Liu, D., Huang, D., Luo, P., & Dai, Y. (2008). New schemes for sharing points on an elliptic curve. *Computers and Mathematics with Applications*, *56*, 1556–1561.
16. MasterCard International & Visa International (1997). Secure electronic transaction specification, Version 1.0. http://www.setco.org/.
17. Menezes, A. J. (1993). *Elliptic Curve Public Key Cryptosystem*. Dordrecht, London: Auburn University, Kluwer Academic.
18. Menezes, A., Okamoto, T., & Vanstone, S. (1993). Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, *39*, 1639–1646.
19. Miller, V. S. (1986). Use of elliptic curve in cryptography. In *Lecture notes in computer sciences: Vol. 218. Advances in Cryptography. Proceedings of CRYPTO'85* (pp. 417–426). New York: Springer.
20. Morales-Sandoval, M., Feregrino-Uribe, C., Cumplido, R., & Algredo-Badillo, I. (2009). An area/performance trade-off analysis of a $GF(2^m)$ multiplier architecture for Elliptic Curve Cryptography. *Computers and Engineering*, *35*, 54–58.
21. Olorunfemi, T. O. S., Alese, B. K., Falaki, S. O., & Fajuyigbe, O. (2007). Implementation of elliptic curve signature algorithm. *Journal of Software Engineering*, *1*(1), 1–12.
22. Ramsdell, B. (1999). S/MIME version 3 message specification. *RFC 2633*.
23. Rivest, R. L., Shamir, A., & Adleman, L. M. (1978). A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, *21*, 120–126.
24. Shi, Z. J., & Yan, H. (2008). Software implementations of Elliptic Curve Cryptography. *International Journal of Network Security*, *7*(1), 141–150.
25. Smart, N. P. (1999). The discrete logarithm problem on elliptic curves of trace one. *Journal of Cryptology*, *12*(3), 193–196.

26. Talbot, J., & Welsh, D. (2006). *Complexity and cryptography*. Cambridge: Cambridge University Press. ISBN978-0-521-85231-9.
27. Vijayalakshmi, V., & Palanivelu, T. G. (2008). Secure localization using Elliptic Curve Cryptography in wireless sensor networks. *International Journal of Computer Sciences and Network Security*, *8*(6), 255–261.
28. Vivek, K., Vivek, S. A., & Ramesh, S. (2008). Elliptic Curve Cryptography. *ACM Ubiquity*, *9*(20), 1–8.
29. Lin, X., Wong, J. W., & Kou, W. (2000). *Performance analysis of secure web server based on SSL* (pp. 249–261). Berlin, Heidelberg: Springer.
30. Ylonen, T., Kivinen, T., Saarinen, M., Rinne, T., & Lehtinen, S. (2003). SSH protocol architecture. IETF Internet.
31. Zona Research (2001). The need for speed II. *Zona Market Bulletin*, (5), 4–8.

O.R. Vincent is a Lecturer in the Department of Computer Science, University of Agriculture, Abeokuta. She obtained a B.Sc. degree in Mathematical Sciences (Computer Science Option) and M.Sc. in Computer Science from the University of Agriculture, Abeokuta in 2000 and 2005 respectively. She is currently on her Ph.D. and studies now with the Computational Intelligence Group, at the Institute of Informatics, Clausthal University of Technology, Germany; where she carries out research on Mobile Agents for E-Commerce. Her research interest include: Images and Vision, Knowledge Management, Computational Complexity, E-commerce, Agents and Mobile Agents. She is a member of Nigeria Computer Society and has published in notable International and local Journals.

O. Folorunso is a Senior Lecturer in the Department of Computer Science, University of Agriculture, Abeokuta. He obtained a B.Sc. degree in Mathematical Sciences from the University of Agriculture, Abeokuta in 1992, M.Sc. in Computer Science from University of Lagos in 1997 and a Ph.D. in Computer Science in 2003 from the university of Agriculture, Abeokuta. His research interest includes Adoption of Information Systems strategies, Human Computer Interaction (HCI), Knowledge Management, Image Processing and Computational Intelligence. He is a member of Nigeria Computer Society and Computer Professional Registration Council of Nigeria. He has published in reputable international and local Journals.

A.D. Akinde The Biography of Prof. Ayodele David Akinde is not available.