RESEARCH ARTICLE OPEN ACCESS

A Closer Look at SMS-Based E-Banking Services Using Elliptic Curves

Ranbir Soram[1], Memeta Khomdram[2], Sonamani Takhellambam[1]
[1]Manipur Institute of Technology, Takyelpat, Imphal -795004, India
[2]National Institute of Electronics and Information Technology, Akampat, Imphal-795001, India

ABSTRACT
With the ushering in of cellphone technology in the country, many financial institutions have launched SMS-based E-banking Services. However, the transmission of SMS in cellphone networks is not secure as the message is sent in plaintext form. So, the content of the message may be exposed to anybody. As SMS-based E-banking Services have become so popular in our daily life, there is a great demand from the users to implement them in a secure environment. Therefore it is desirable to make SMS-based e-banking Services secure by additional encryption. In this paper, we study SMS-based E-Banking Services using Elliptic Curves. We also give the benefits of using Elliptic Curves over RSA in SMS-based E-Banking Services.
Keywords – Elliptic Curve, SMS, Encryption, Decryption, E- Banking

I. INTRODUCTION

SMS-based E-banking Service is a new service provided by most financial institutions i.e., Banks in India. It is a term used for performing balance checks, account transactions, payments through a mobile device such as a mobile phone. The SMS-based E-banking Service is based on the exchange of SMS messages between customers and the bank. SMS-based E-banking Service has seen an explosive growth in the country. The main reason for the popularity of SMS-based E-banking Service over Internet Banking Service is that it enables 'Anywhere and Anytime Banking'. Customers now don't need access to a computer terminal to access their banks, they can now do so on the go – when they are waiting for their bus to board, when they are traveling or when they are waiting for their orders to come through in a restaurant. There are two methods of SMS-based e-banking services widely used today; they are the Push and Pull message services [1].

Push service is the message that the bank sends out to a customer's mobile phone, without the customer initiating a request for the information. An example of push message could be a withdrawal alert, which alerts the user when a withrawal is made from his account [1].

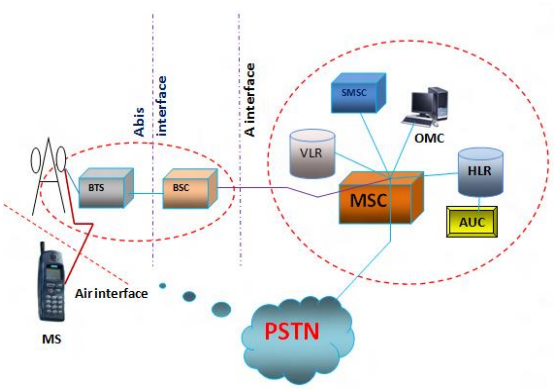
Pull message service is a request initiated by the customer, using a mobile phone, for obtaining information or performing a transaction in the bank account.This is a full duplex communication system where a user sends a request to the bank and the bank replies with the information sought by the user. An example of pull SMS message is an account balance enquiry made by a user.

The other way to classify the SMS-based e-banking services, by the nature of the service, gives us two kinds of services – Enquiry based and Transaction based. A request for your bank statement is an enquiry based service and a request for your fund transfer to some other account is a transaction based service. Transaction based services are also differentiated from enquiry based services in the sense that they require additional security across the channel from the mobile phone to the banks data servers.

II. CELL PHONE ARCHITECTURE

Many countries with mobile communication services use **G**lobal **S**ystem for **M**obile Communication (GSM) architecture to network their mobile connections. GSM network was initially designed to be used for voice communication. As the usage of mobile phone increases, people begin to use their mobile phones for additional means of data transmissions. The most popular type of data transmission is Short Message Service (SMS). The SMS is a GSM technology that allows exchange of text messages up to 160 characters of user data, which can comprise of words or number or an alphanumeric combination among mobile phones through the Short Message Service Center (SMSC) of the particular network operator. The relative ease of the use of SMS makes it the most wanted means of communication among mobile users. SMS protocol uses control channel instead of traffic channel. The control channel is also used for calls initiation. Therefore, if the control channel is flooded with SMS messages, then there would be no chance for the call initiation signal to get through and it causes a denial of service attack. SMS messages are sent asynchronously. The SMS processing computers usually run on corporate servers that are connected to the GSM network through specialized routers and gateways connected to the

SMS centers of the mobile operators. When a message is submitted for sending, the service provider will keep the sending message in its buffer until the message is delivered to the destined mobile phone. GSM is a globally accepted popular standard for digital mobile phones in the world. It stands for Global System for Mobile communication (initially **G**roup **S**pecial **M**obile). It is used by over 5 billion people across more than 212 countries and territories.



MS= Mobile Station
BSC= Base Station Controller
HLR= Home Location Register
AUC =Authentication
BTS= Base Transceiver Station
MSC=Mobile Switching Center
VLR= Visitor Location Register
OMC=Operations and Maintenance

The solid lines show how the communication signals transfer between the essential components.

Fig 1: Cell Phone Network Architecture

GSM's signaling and speech channels are digital, and thus is considered a second generation (2G) mobile phone system. In Fig. 1 above, we illustrate an overview of the GSM architecture and it consists mainly of the following functional parts:

MSC:- The mobile switching center is the core switching element in the network. It controls calls to and from other telephone systems. It also performs such functions as toll ticketing, network interfacing, common channel signaling. A GSM network has one or more MSCs, geographically distributed.

VLR:- The visitor location register is another database containing temporary data for subscribers registered in an MSC. This information is needed by MSC to serve the visiting subscribers. If a mobile visits a new area, the VLR there will request data about this visitor from the HLR. Every MSC contains a VLR. Although MSC and VLR are individually addressable, they are always contained in one integrated node.

HLR:- The home location register is a database that contains permanent information for each subscriber of the network. The subscriber's address, service profile and activity status are in HLR. A GSM subscriber is normally associated with one particular HLR.

AUC:- The authentication center provides authenticated services like identifying authorized users and also ensures the confidentiality of each call.

BSC:- The base station controller provides all the control functions and physical links between the BTS and MSC, and controls handoffs, radio frequency, and power levels in BTS.

BTS:- The base transceiver station is a radio equipment containing both transceiver and antenna. It is responsible for radio signal transmission and reception.

MS:- The mobile station is made up of two entities:

ME:-The Mobile Equipment is produced by many different manufacturers who obtained approval from standardization body. It is uniquely identified by an IMEI (International Mobile Equipment Identity).

SIM:- The Subscriber Identity Module is a smart card containing the International Mobile Subscriber Identity (IMSI). It allows user to send and receive calls and receive other subscribed services. It is protected by a password or PIN and can be moved from phone to phone.

III. SMS ARCHITECTURE AND PROTOCOL

Even if we are not talking on our cell phones, the phones are constantly sending and receiving information. It is talking to the mobile tower over a link called a control channel. The reason for this perpetual exchange of information is that the cell phone system knows which cell our phones are in, and so that our phones can change cells as we move around. Every often, our phones and the tower will exchange a packet of data so that they (mobile tower + cell phones) know that everything is alright. Our phones also use the control channels for call setup. When someone tries to call you, the mobile tower sends your phone a message over the control channel that tells your phone to play its ringtone. The mobile tower also gives your phone a pair of voice channel frequencies to use for the call. The control channel also provides the pathway for SMS messages. When Alice sends Bob an SMS message, the message flows through the SMS Gateway, the SMSC, then to the tower, and the tower sends the message to Bob's phone as a little packet of data on the control channel.

The SMS Gateway is located in the application layer. Please refer to Fig. 2 given below. When sending an SMS message, the software creates

protocol data units (PDUs) transported by the transport layer. SMS Gateway decodes this PDU and makes the message readable for computer programs and computer users. To understand how the SMS travels from the mobile phone to the SMSC, look at Fig. 3 given below.

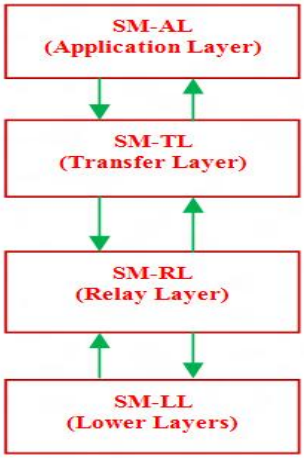


Fig 2: Protocol layers in a GSM network

In fig.2 we can see which protocols are used and which GSM network entities take place in the communication process. As we can see, the mobile station transmits the SMS message to the BTS through a wireless link. Then the message goes through the backbone network of the service provider. The MSC, the HLR and, optionally, the VLR are used to find out the appropriate SMSC which will store and forward the message when the receiving party becomes available. The SMS would be automatically stored on the handset and be available to anyone that looks at the user's phone. As can be seen, there are many points of exposure.

IV. THE STATE OF SMS

SMS was created in the late 1980s to work with GSM technology. The Norwegian engineers who invented it wanted a very simple messaging system that worked when users' mobile phones were turned off or out of signal range. Internet sources say that the first SMS message was sent in the UK in the early 1990s.

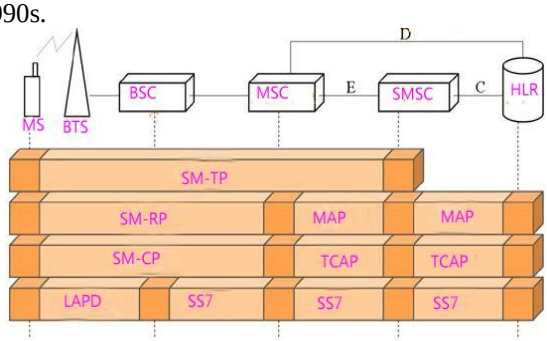


Fig 3: SMS architecture and protocol stack

The use of SMS in E-banking is needed not because it is secure and convenient to use but because alternatives are not available or are costly to implement. There are certain guidelines issued by the Central Bank of the country to be followed by all financial institutions in the country. It has made two-factor authentication mandatory for all E-banking Services in the country. Two-factor authentication adds a layer of protection to the standard password method of online identification. "Two Factor Authorization" allows for "One Time Passwords" for E-Banking transaction authentication to be delivered via SMS to your Mobile Phone. Experts are quick to point out the shortcomings of two-factor authentication: it usually requires a USB token, phone, or other device that's easy to lose; and it is subject to man-in-the-middle attacks as the SMS itself is not very secure. But, still, for online banking and other Web transactions, two-factor authentication is the most practical protection available. ICICI bank is the first bank in the country to have introduced E-banking for a limited range of services such as access to account information, correspondence and, recently, funds transfer between its branches.

V. SECURITY PROBLEM WITH SMS

The technical specifications for SMS are given in ETSI TS 03.48. As the initial idea for SMS usage was intended for the subscribers to send non-sensitive text messages across the open GSM network, many security concerns such as SMS text encryption and mutual authentication were omitted during the design. of GSM architecture. In practical use, SMS messages are not encrypted during transmission. A cyclic redundancy check is provided during SMS transmission to ensure that the short messages do not get corrupted. Each short message has a validity period whereby temporary storage is provided by the SMSC if the SMS message cannot be delivered to the recipient successfully. The SMSC will delete stored SMS messages if they cannot deliver a message within the validity period. Since encryption is not applied to short message transmission by default, messages could be intercepted and snooped during transmission. In addition, SMS messages are stored as plain text by the SMSC before they are successfully delivered to the intended recipient. These messages could be viewed or amended by users in the SMSC who have access to the messaging system.

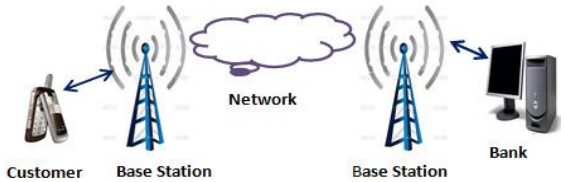


Fig 4: SMS-based E-banking

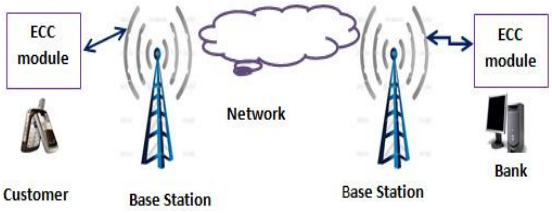


Fig 5: SMS-based E-banking using Elliptic Curves

VI. TRADITIONAL SMS-BASED E-BANKING

The traditional SMS-based e-banking security is discussed at length in [1]. Presently, customers have to submit the registration form by giving their cell phone number, account number and transaction details. The customer can receive his account balance and transactions only when the request is received from the cell phone number registered with the bank and duly authenticated by the 6-digit number as in the case OnlineSBI, India. A customer would initiate a transaction by sending SMS to the bank using the bank's SMS short code as a terminating address. The SMS would be automatically stored on the handset and be available to anyone that looks at the customer's phone. Data being carried across the mobile network jumps from one base station to the next, which means that the chain of encrypted communication between the customer and the bank is broken. So, current mobile banking services offered by banks are not secure enough to protect confidential data.

Table 1: The public key (**n**) of Vijaya Bank

24957916309063346482186250739143982599680313249857490164204211355601103291059839143047307274492781547590559782399419127443278028597408773514989409720796105182979801620931403761314620004350719952971693865357904574335916920535716233685120162717615822133072086614835702181971438623800983233171787349226625393933084267254349815824936234010521961761188302408029610752543199688518109993680702036643868958960771343594328130681755557105708658329666726369741990675507074447081472348770639781446555807722019418153108888869532543835904180812969227630710646250603429539555160067537916211315198489227577791495495950459230659895 57

VII. PROBLEM OF RSA

One of the main concerns of RSA is the demand for larger keys in today's cryptographic algorithms. An RSA key length of 1024 bits is used in web site logins but for high-security applications such as online financial fund transfers or for data that need to remain confidential for more than a few years; a 2048-bit key is recommended. The huge number given in table 1 is the public key of Vijaya Bank. Life is changing at a very fast pace, computers have become more and more powerful and, therefore, security requirements constantly change resulting in the demand for higher keys. What is perfectly acceptable and more than enough today may not be sufficient enough tomorrow. With every doubling of the RSA key length, decryption is about 8 times slower; encryption is slower by a factor of 4.The size of cipher text also become huge considerably. But it's usually the speed of decryption that we're more worried about because that's the part that takes place on the server and the decryption is very much slower than encryption, because the decryption exponent is huge whereas the encryption exponent is typically small. Even if we are able to sacrifice some amount of CPU time for free for decryption, it leaves us another problem- an attacker can consume a few seconds of CPU time on our server by firing some random data at it making our server down. This is the main problem of RSA. With suitable restrictions on the rate of login attempts (and thus decryptions) per remote client, we may protect a "CPU burn" attack.

VIII. ECC BANKING MODULE

In order to perform SMS-based E-Banking in a secure environment, a system that would provide security at the satisfaction of the users is proposed in [1]. This system is called the ECC module. This ECC module receives the text messages from the senders and processes them and sends the output back to the recipients as and when required. This ECC module provides encryption and decryption of user data using Elliptic Curve Public Key Cryptography. The block diagram of the new system is given in fig. 5 given above.

The silent features of the new system are enumerated below:-

(i) A strong cryptographic technology called the Elliptic Curve Cryptography, instead of traditional RSA, is used. We give a quick discussion on Elliptic Curve Cryptography is Section X.

(ii) Two ECC modules are used. One module is in the handset of sender and another module is in the bank.

(iii) In order to provide message authentication, message integrity and non-repudiation of message, digital signature using ECC technology may optionally be used in each of the module. Unfortunately, SMS exchange in GSM technology does not provide these.

The ECC module in the handset is one of the design issues in this system. There are two possible solutions for this issue. One solution is to house the ECC module as a SIM Application Toolkit (STK). The STK is a set of commands which enables the SIM to initiate actions which can be used for various purposes. STK has been deployed by many mobile operators around the world for many applications, often where a menu-based approach is required. STK

has been deployed on the largest number of mobile devices. Housing the ECC module as an STK application can be done in two ways- either the SIM must be returned and exchanged for a new one or the ECC module must be delivered over-the-air (OTA) using specialized and optional SIM features. The first method may be inconvenient to most customers.

Another option to house the ECC module is in the handset itself. This is not a big issue as most handsets come with a good amount of space and memory. So, a new handset can come with pre-built ECC module whereas existing handsets may be reprogrammed to house the ECC module.

IX. ELLIPTIC CURVE

Elliptic curves are a specific class of algebraic curves. The "Weierstrass form" of an elliptic curve equation is [2],[4]:-

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

The constant $a_1,\ a_2,\ a_3,\ a_4,\ a_6$ and the variables x, y can be complex, real, integers, polynomials, or even any other field elements. But in practice we must specify which field, F, these constants and the variables, x, y belong to and $\Delta \neq 0$, where Δ is the discriminant of E and is defined as follows [2,4]:-

$$\Delta = -d_2^2 d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6$$
$$d_2 = a_1^2 + 4a_2$$
$$d_4 = 2a_4 + a_1a_3$$
$$d_6 = a_3^2 + 4a_6$$
$$d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2$$

We say that E is defined over K when the coefficients $a_1,\ a_2,\ a_3,\ a_4,\ a_6$ (and of course, the variables x and y) of the equations come from the elements of the field K. So, we sometimes write $E(K)$ to emphasize that E is defined over K, and K is called the underlying field.

A. ELLIPTIC CURVE OVER A PRIME GALOIS FIELD

An elliptic group over a prime Galois Field uses a special elliptic curve of the form

$$y^2 \pmod p = x^3 + ax + b \pmod p$$

where $a, b \in GF(p), 0 \le x \le p$ and $-16(4a^3 + 27b^2) \bmod p \neq 0$. The constants ***a*** and ***b*** are non-negative integers smaller than the prime p. The condition that $-16(4a^3 + 27b^2) \bmod p \neq 0$ implies that the curve has no "singular points" [2],[4].

B. GROUP LAW

The mathematical property that makes elliptic curves useful for cryptography is simply that if we take two distinct points on the curve, then the chord joining them intercepts the curve in a third point for because we have a cubic curve. If we then reflect that point in the x-axis we get another point on the curve as the curve is symmetric about the x-axis. This is the "sum" of the first two points. Together with this addition operation, the set of points $E(K)$ forms an abelian group with **0** serving as its identity [2],[4]. It is this group that is used in the construction of elliptic curve cryptographic systems.

Group law for $y^2 = x^3 + ax + b$ over $GF(p)$.

(1) **Identity:** $P + 0 = 0 + P = P$ **for all** $P \in E(K)$.

(2) Negative: If $P = (x, y) \in E(K)$, then $(x, y) + (x, -y) = 0$. The point $(x, -y)$ is denoted by -P and is called the negative of P; note that -P is indeed a point in $E(K)$. Also, $-0 = 0$.

(3) Point addition: Let $P = (x_1, y_1) \in E(K)$ and $Q = (x_2, y_2) \in E(K)$ where $P \neq \pm Q$ .Then $P + Q = R(x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2,\ y_3 = \lambda(x_1 - x_3) - y_1$ and $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$.

(4) Point doubling: Let $P = (x_1, y_1) \in E(K)$, where $P \neq \pm P$. Then $2P = R(x_3, y_3)$, where $x_3 = \lambda^2 - 2x_1,\ y_3 = \lambda(x_1 - x_3) - y_1$ and $\lambda = \frac{3x_1^2 + a}{2y_1}$.

The geometrical interpretation of the above group law is given here. Let's take a point $P = (x, y)$. The formula for finding $-P$ is $-P = (x, -y)$ as shown in the fig. 6.

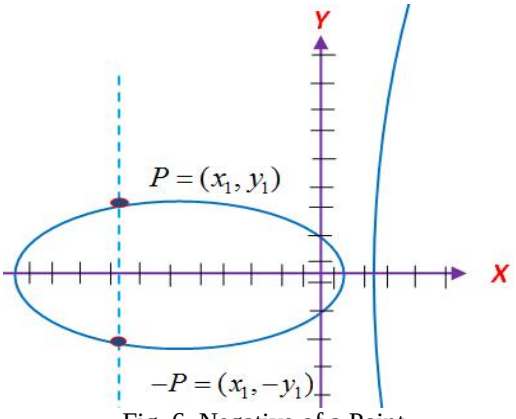


Fig. 6. Negative of a Point

We can define the addition of any two points on an elliptic curve by drawing a line between the two points and finding the point at which the line intersects

the curve. The negative of the intersection point is defined as the "elliptic sum" of the two points and is shown in fig. 7.

Mathematically we write:

$$R = P + Q.$$

This "addition" satisfies all the usual algebraic properties that we associate with integers, provided we define a single additional point "the point at infinity", which plays

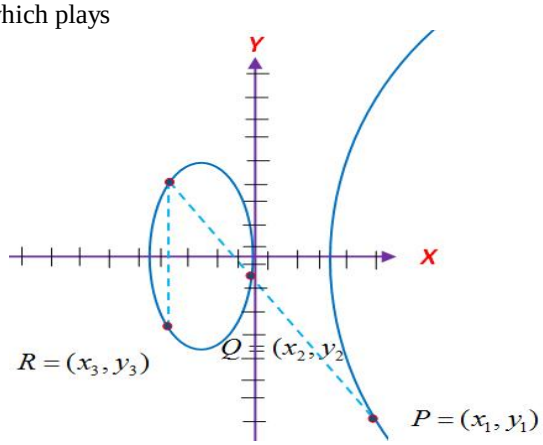


Fig. 7. Addition of two Points

the role of **0** in the integers. In mathematical terms, we can define a finite additive abelian group on the points of the curve, with the zero being the point at infinity. If $P = (x_1, y_1)$, then the double of *P*, denoted by, $R = (x_3, y_3)$, is defined as follows. First draw the tangent line to the elliptic curve at *P*. This line intersects the elliptic curve in a second point. Then *R* is the reflection of this point in the x –axis. This is depicted in fig. 7.

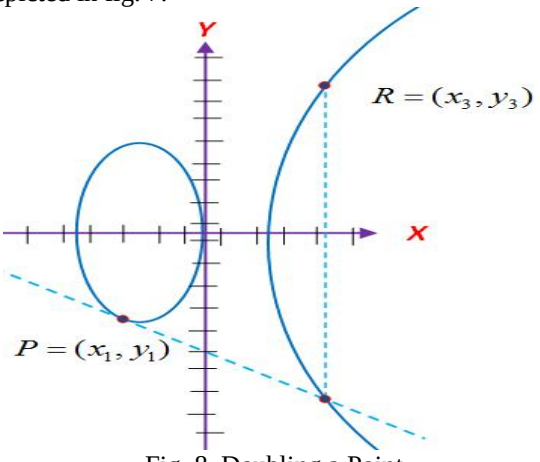


Fig. 8. Doubling a Point

We can extend this idea to define $P + P + P = 3P$, and extending this idea further, we can define $P + P + P + ... + k$ times $= kP$, for any integer *k*, and hence define the order of P, being the smallest integer k such that $kP = 0$, where 0 denotes the point at infinity. Fig. 9 shows some multiples of $P = (-1, -2)$ on the curve $y^2 = x^3 - 5x$.

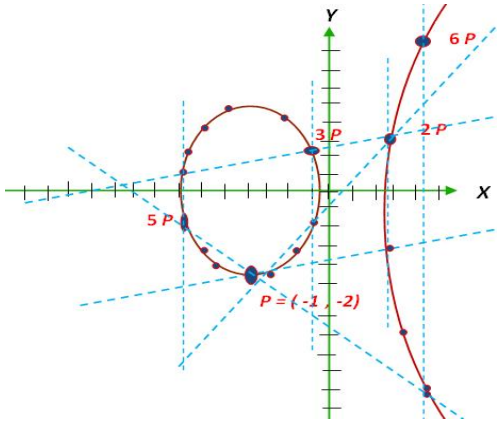


Fig. 9. Some Multiples of $P = (-1, -2)$.

C. ELLIPTIC CURVE OVER $GF(2^n)$.

Now it is time to have a look at Elliptic Curves over $GF(2^n)$. That means our constants are either polynomial or normal basis numbers. We cannot use the simplified version of equation which we used for integer numbers.

Experts in Elliptic Curve Cryptography suggest us that we use either of the versions given below:

$$y^2 + xy = x^3 + ax^2 + b \quad (1)$$

$$y^2 + y = x^3 + ax + b \quad (2)$$

Mathematicians call the second form above, equation (2), a "supersingular" curve. These forms of equations can be computed very quickly. However, these curves are unsuitable for cryptography. See [1] for more information..

The curves of equation (1) are called "nonsupersingular" curves. From technical points of view, curves of this form are excellent for cryptographic applications. We must be careful in choosing the coefficients to get more benefits in terms of security. A poor choice can create a curve that is easier for the cryptanalyst to attack. For equation (1) to be valid, *b* must never be 0. However, *a* can be 0. The rules are the same as before: Take any two points on the curve; draw a line between them; and the negative of the third point, which intersects both the curve and the line, is the "sum" of the first two points. Here we give the group laws of the first form of the curve [1], [2].

Group law for $y^2 + xy = x^3 + ax^2 + b$ **over** $GF(2^n)$

I. Identity: $P + 0 = 0 + P = P$ for all $P \in E$.

II. Negative: If $P = (x, y) \in E$, then $(x, y) + (x, x + y) = 0$. The point $(x, x + y)$ is denoted by -P and is called the negative of P; note that -P is indeed a point in E. Also, $-0 = 0$.

III. Point addition: Let $P = (x_1, y_1) \in E$ and

$Q=(x_2,y_2)\in E$ where $P\neq\pm Q$ .Then $P+Q=R(x_3,y_3)$, where $x_3=\lambda^2+\lambda+x_1+x_2+a$ and $y_3=\lambda(x_1+x_3)+x_3+y_1$ with

$$\lambda=\frac{y_2+y_1}{x_2+x_1}$$

IV. Point doubling: Let $P=(x_1,y_1)\in E,$ where $P\neq -P$. Then $2P=R=(x_3,y_3),$ where $x_3=\lambda^2+\lambda+a$ and $y_3=x_1^2+\lambda x_3+x_3$ with

$$\lambda=x_1+\frac{y_1}{x_1}$$

TABLE 4. POSSIBLE VALUES OF g's

0	*000*	$g^3=g+1$	*011*
1	*001*	$g^4=g^2+g$	*110*
g	*010*	$g^5=g^2+g+1$	*111*
g^2	*100*	$g^6=g^2+1$	*101*

Let us take an elliptic curve [11] $y^2+xy=x^3+g^3x^2+1$ over $GF(2^3)$ under the irreducible polynomial $f(x)=x^3+x+1.$ Here the generator, g, satisfies the relation $g^3+g+1=0$ or $g^3=g+1$ as the arithmetic is over $GF(2).$ The following table 4 shows the values of $g's$ and the points on the curve are given in table 5.

TABLE 5. POINTS ON THE GIVEN CURVE

0	$(0,1)$	$(g^2,1)$	(g^2,g^6)
(g^3,g^2)	(g^3,g^5)	$(g^5,1)$	(g^5,g^4)
(g^6,g)	(g^6,g^5)		

Let $P=(0,1)$ and $Q=(g^2,1).$ We have $P+Q=R=(x_3,y_3)$ is computed as follows.

$$\lambda=\frac{1+1}{g^2+0}=0$$

$x_3=\lambda^2+\lambda+x_1+x_2+a=0+0+0+g^2+g^3=g^5.$

and

$y_3=\lambda(x_1+x_3)+x_3+y_1=0(0+g^5)+g^5+1$

$=g^5+1=g^2+g=g^4.$

So, $R=(g^5,g^4)=(111,110).$

Again take $P=(g^2,1).\quad 2P=P+P=R(x_3,y_3).$

$$\lambda=g^2+\frac{1}{g^2}=g^2+g^5=g+1=g^3$$

$x_3=\lambda^2+\lambda+a=g^6+g^3+g^3=g^6.$

and

$y_3=x_1^2+\lambda x_3+x_3$

$=g^4+g^9+g^6=g^4+g^2+(g^2+1)$

$=g^4+1=(g^2+g)+1=g^5.$

Therefore, $R=(x_3,y_3)=(g^6,g^5)=(101,111).$

D. HASSE THEOREM AND POINT COUNTING

Let *E* be an elliptic curve defined over F_q. The number of points in $E(F_q)$, denoted by $\#E(F_q)$, is called the *order* of *E* over F_q. Then Hasse's theorem says that the order of $E(F_q)$ satisfies the inequality [3]

$$q+1-2\sqrt{q}\le\#E(F_q)\le q+1+2\sqrt{q}.$$

An alternate formulation of Hasse's theorem is the following: if *E* is defined over F_q, then $\#E(F_q)=q+1-t$ where $|t|\le 2\sqrt{q}$; *t* is called the *trace* of *E* over F_q. Since $2\sqrt{q}$ is small relative to *q*, we have $\#E(F_q)\approx q$.

There are several methods presently known that can quickly determine the order of $E(F_q)$. Unfortunately none of them is effective once q is very large. An alternative approach is to use the order of certain points in $E(F_q).$ Since $E(F_q)$ is a group, and then the order of any point in $E(F_q)$ must divide $\left|E(F_q)\right|$, by Lagrange's theorem. In Hasse's

theorem, we know that $|E(F_q)|$ is bounded in an interval of length $4\sqrt{q}$. If we can find a point in $E(F_q)$ of order $m > 4\sqrt{q}$, then there will be only one multiple of m lying in that interval, which must be $|E(F_q)|$. For example, let E be the elliptic curve $y^2 = x^3 - 10x + 21$ over $GF(557)$. It can be shown that the point (2, 3) has order 189. Hasse's theorem says that

$$557 + 1 - 2\sqrt{557} \le |E(F_{557})| \le 557 + 1 + 2\sqrt{557}$$

$$\text{i.e, } 511 \le |E(F_{557})| \le 605$$

But the only multiple of 189 in this interval is 3 as $3X189 = 576$. Hence, $|E(F_{557})| = 567$.

E. SUPERSINGULAR CURVES

Elliptic curves defined over a finite field are of two types. Most are what are called ordinary or non-supersingular curves, but a small number are supersingular[1]. As mentioned in [3], the order or cardinality of an elliptic curve is $\#E(F_q) = q + 1 - t$, where $|t| \le 2\sqrt{q}$. Let *p* be the characteristic of F_q. An elliptic curve *E* defined over F_q is supersingular if *p* divides *t*, where *t* is the trace. If *p* does not divide *t*, then *E* is non-supersingular [2]. The problem with the supersingular elliptic curve is that the ECDLP in an elliptic curve E defined over a field F_q can be reduced to the ordinary DLP in the multiplicative group of some finite extension field of $F_q k$ for some $k \ge 1$. It follows that the reduction of ECDLP to ordinary DLP can be solved in a sub-exponential time, thus, compromising security of the system. To ensure that the reduction does not apply to a particular curve, one need to make sure that n, the order of the point P, does that divide $q^k - 1$ for small k.

F. AN IMPORTANT THEOREM

Let *E* be an elliptic curve defined over F_q. Then $E(F_q)$ is isomorphic to $Z_{n1} \oplus Z_{n2}$ where n_1 and n_2 are uniquely determined positive integers such that n_2 divides both n_1 and $q - 1$. Note that $\#E(F_q) = n_1 n_2$. If $n_2 = 1$, then $E(F_q)$ is a cyclic group. If $n_2 > 1$, then $E(F_q)$ is said to have rank *2*. If n_2 is a small integer (e.g., *n* = 2 ,3 or 4), we sometimes say that $E(F_q)$ is almost cyclic [1],[2],[11]. Since n_2 divides n_1 and $q - 1$, one expects that $E(F_q)$ is cyclic or almost cyclic for most elliptic curves *E* over F_q.

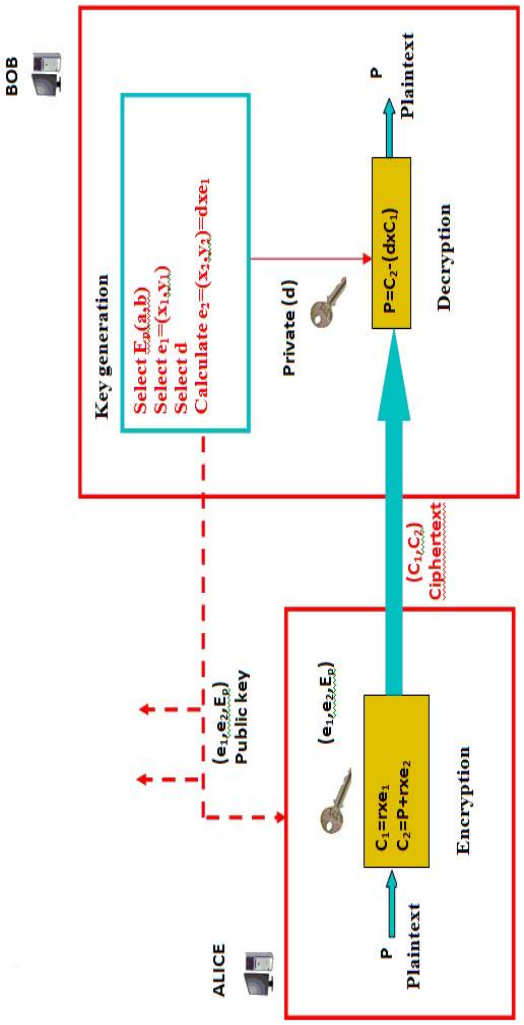


Fig. 10: ECC Encryption and Decryption block diagram

X. ECC ENCRYPTION AND DECRYPTION

Elliptic Curve Cryptography has been used to encrypt plaintext messages, **M**, into ciphertexts, C, and decrypt ciphertexts into plaintext messages as in fig. 10. The plaintext message **M** is to be encoded into a point P_m from the finite set of points in the elliptic group, $E_p(a,b)$. We first convert the plaintext message **M** into a sequence of integers (normally a single integer for technical reason) and mapped to a point on the curve.

A. KEY GENERATION

1. Alice and Bob agree on a common domain parameter $D = (q, FR, S, a, b, G, n, h)$ where the generator point $G = (x_g, y_g)$ is carefully chosen.

2. Alice chooses an integer n_a and calculates $P_a = n_a G = (x_a, y_a)$ according to group law.
3. Alice's public key is $P_a = (x_a, y_a)$ and his private key is n_a.
4. Bob also chooses an integer n_b and calculates $P_b = n_b G = (x_b, y_b)$ according to group law.
5. Bob's public key is $P_b = (x_b, y_b)$ and his private key is n_b.

B. ENCRYPTION

Alice wishes to send a message $P_m = (x_m, y_m)$ to Bob. He carries out the following steps.

1. Alice chooses a random number *k*.
2. He calculates $c_1 = kG$ and $c_2 = P_m + kP_b$..
3. Alice sends the $C_m = \{c_1, c_2\}$ as cipher text to Bob.

C. DECRYPTION

Upon receiving the ciphertext pair $C_m = \{c_1, c_2\}$ from Alice, Bob recovers the message as follows:

He multiplies c_1 by his private key n_b and subtracts it from c_2. That is, he calculates

$$c_2 - n_b c_1 = (P_m + kP_b) - n_b(kG) = (P_m + kn_b G) - n_b kG$$
$$= P_m = (x_m, y_m)$$

XI. ELLIPTIC CURVE CRYPTOGRAPHY EXAMPLE

To illustrate Elliptic Curve Cryptography, consider the following elliptic curve [1]:

$$y^2 = x^3 - x + 188 \bmod 751$$

The elliptic curve group generated by the above elliptic curve is $E_{751}(-1,188)$. We have to choose a point as the generator point. In fact any point on the curve can be a generator point. Here, let the generator point be $G = (0,376)$. Then multiples of the generator point G are:

$G = (0,376)$ $\quad 2G = (1,376)$
$3G = (750,375)$ $\quad 4G = (2,373)$
$5G = (188,657)$ $\quad 6G = (6,390)$
$7G = (667,571)$ $\quad 8G = (121,39)$
$9G = (582,736)$ $\quad 10G = (57,332)$
.....
$761G = (565,312)$ $\quad 762G = (328,569)$
$763G = (677,185)$ $\quad 764G = (196,681)$
$765G = (417,320)$ $\quad 766G = (3,370)$
$767G = (1,377)$ $\quad 768G = (0,375)$
$769G = \mathrm{O}$ (point at infinity)

If Alice wants to send to Bob the message M which is encoded as the plaintext point $P_m = (443,253) \in E_{751}(-1,188)$, she must use Bob public key to encrypt it. Suppose that Bob secret key $n_b = 85$, then his public key will be $P_b = d\mathrm{x}G = 85(0,376) = (671,558)$.

Alice selects a random number $k = 113$ and uses Bob's public key $P_b = (671,558)$ to encrypt the message point into the ciphertext pair of points:

$$[c_1, c_2] = [(kG), (P_m + kP_b)]$$
$$= [113\mathrm{x}(0,376), (443,253) + 113(671,558)]$$
$$= [(34,633), (443,253) + (47,416)]$$
$$= [(34,633), (217,606)]$$

Upon receiving the ciphertext pair of points, $[c_1, c_2] = [(34,633), (217,606)]$, Bob uses his private key, $n_b = 85$, to compute the plaintext point, P_m, as follows

$$c_2 - n_b c_1 = (P_m + kP_b) - [n_b(kG)]$$
$$= (217,606) - [85(34,633)]$$
$$= (217,606) - [(47,416)]$$
$$= (217,606) + [(47,-416)] \text{ since } -P = (x_1, -y_1)$$
$$= (217,606) + [(47,335)] \text{ since } -416 \equiv 335 \bmod 751$$
$$= (443,253)$$

and then maps the plaintext point $P_m = (443,253)$ back into the original plaintext message M.

XII. ENCODING PLAINTEXT AS POINTS ON AN ELLIPTIC CURVE

Suppose E is an elliptic curve given by $y^2 = x^3 + ax + b$ over $GF(p)$. Here we give a technique to embed the message in the x-coordinate of a point on the curve. Let k be a large enough integer such that the probability of failing to encode a plaintext message m is 1 out of 2^K. In practice $K = 30$ or at worse $K = 50$ should be sufficient enough.

Suppose that our message m is integer satisfying $(m+1)K < p$ and message *m* will be represented by $x = mK + i$ where $0 \le i \le K$. For each $i = 0,\ 1,\ 2,\ 3,\ldots,\ K-1$; compute x_i and also the right side of the equation

$$y_i^2 = f(x_i) = x_i^3 + ax_i + b,$$

and try to find a square root of $f(x_i)$. If we find a y_i such that $y_i^2 = f(x_i)$, we take $P_m = (x_i, y_i)$. If it turns out that $f(x_i)$ is a non-square, then increment *i* by 1 and try again with the corresponding x_i ,provided we find an x_i for which $f(x_i)$ is a square before i gets bigger than k.To recover m from (x_i, y_i), simply compute $\left[\frac{x_i}{K}\right]$ (i.e., the greatest integer less than or equal to $\frac{x_i}{K}$).Since $f(x_i)$ is a square for approximately 50% of all x_i, there is only about a $\frac{1}{2^k}$ probability that this method will fail to produce a point P_m.

As an example, we use an elliptic curve $E: y^2 = x^3 + 2x + 7$ defined over $GF(179)$. Assume we are satisfied with a failure rate of $1/2^{10}$, and then we may take $K = 10$. Since $(mK + K) < 179$, we need $0 \le m \le 16$. Suppose our message is m=5. Here the possible choices for *x* are *50, 51,...,59.* For *x=51*, we get $x^3 + 2x + 7 \equiv 121 (\text{mod } 179), 11^2 \equiv 121 (\text{mod } 179).$

Hence $P_m = (51,11)$. The message m can be recovered as $\left[\frac{51}{10}\right] = 5.$

XIII. SECURITY OF ECC

Let E be an elliptic curve defined over a finite field and let, P be a point (called base point) on E of order *n* and *k* is a scalar. Calculating the point $Q = kP$ from P is very easy and $Q = kP$ can be computed by repeated point additions of P. However, it is very hard to determine the value of k knowing the two points: kP and P. This lead leads to the definition of Elliptic Curve Logarithm Problem (ECDLP), which is defined as: "Given a base point *P* and the point $Q = kP$, lying on the curve, find the value of scalar k". The integer k is called the Elliptic Curve Discrete Logarithm of Q to the base P, denoted as $k = \log_P Q$.

As an example consider the group $E_{23}(9,17)$,defined by the elliptic curve equation $y^2 \ (mod\ 23) = \left(x^3 + 9x + 17\right) mod\ 23$. Let us find out the Elliptic Curve Discrete Logarithm **k** of **Q**=(4,5) to the base **P**=(16,5). The brute-force method is to compute multiples of **P** until **Q** is found.

Thus,

$P = (16,5);\ 2P = (20,20);$

$3P = (14,14);\ 4P = (19,20);$

$5P = (13,10)\ ;\ 6P = (7,3)\ ;$

$7P = (8,7)\ ;\ 9P = (4,5).$

Because 9**P**=(4,5)=**Q**, the discrete logarithm of Q=(4,5) to the base P=(16,5) is **k**=9. In a real application, **k** would be so large as to make the brute-force method almost impossible.

XIV. DEVELOPMENT OF API

The location of encryption is very important and it is discussed at length in [1]. There are two general approaches for encryption in computer network: link encryption and end-to-end encryption. With link encryption, each vulnerable communications link is equipped on both ends with an encryption device. With end-to-end encryption, the encryption process is carried out at the two ends of the system.

Fig. 11. Network Layer Encryption. In this case, data and more headers are encrypted. Here **H1**, **H2** and **H3** are headers.

Fig. 12. Transport Layer Encryption. In this case, data and less headers are encrypted. Here **H1**, **H2** and **H3** are headers.

Fig. 13. Application Layer Encryption. In this case, only data portion is encrypted. Here **H1**, **H2** and **H3** are headers.

For end-to-end encryption, several choices are possible for the placement of the encryption function. We can place it in the network layer or transport layer. The user data portion and some headers of all frames

are encrypted. See fig. 11 and 12 given above. However, if the message passes through a node or gateway, the line connection is terminated and a new connection is opened for the next hop or node. Thus, encrypted data (our message + some headers) are decrypted at the gateway. Before transmission to the next node, it is again encrypted. So, our data are not secure at the intermediate nodes like gateways.

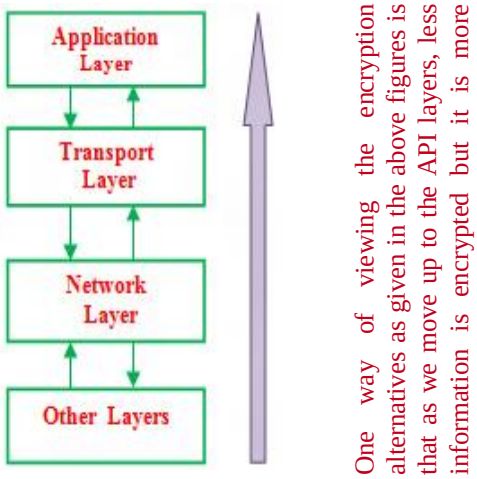


Fig. 14: Encryption Alternatives

For application that has a store-and-forward capacity, the only place to achieve end-to-end encryption is at the application layer. A drawback of application layer encryption is that the number of entities increases considerably.

With application level encryption, only the user data portion of a segment is encrypted. See Fig. 13 given above. The headers are all clear and visible. So, no decryption and encryption of data take place at the intermediate nodes or gateways.

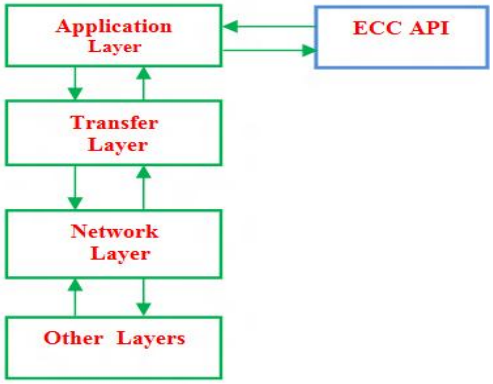


Fig. 15: Where the API fits in the overall network

To get maximum security, it is suggested that the ECC API be used in the application layer.

XV. BENEFITS OF ECC OVER RSA

Elliptic Curve Cryptography offers many advantages over RSA in many dimensions. One advantage of Elliptic Curve Cryptography over RSA system is that smaller parameters still with same security level cane be used. The advantages that we gain from smaller parameters include faster computations and smaller certificates.

A. PUBLIC KEY SIZE OF RSA AND ECC

An RSA public key pair consists of an ordered pair (n,e) where n is a composite number, called the modulus, and e is the public exponent. In a 1024-bit RSA system, n will have 1024 bits. For some technical reasons, we normally choose the public exponent as $e=2^{16}+1(=65537)$. Thus, an RSA public key would require 128 bytes for the modulus and (2+1=) 3 bytes for the public exponent. The total size is then 131 bytes.

Table 2: Relative public key sizes of RSA and ECC

Security Level	RSA	ECC
80 bit	1024	160
112 bit	2048	224
128 bit	3072	256
140 bit	4096	280
192 bit	7680	384
300 bit	21000	600

An ECC public key consists of a point on the elliptic curve. Each point is represented by an ordered pair of element (x, y). For a 192-bit elliptic curve, the public key is then represented by two 24-byte numbers, giving a total key size of 48 bytes.

To reduce the size of the ECC public keys we go for point compression. Using point compression, a public key could be represented by using one 192-bit value and one additional bit (truly speaking 1 byte). This would then require (24+1=) 25 bytes.

As can be referred from above, ECC provides a significant reduction in public key size. This reduction is very much essential in many constrained environments where large public keys are not possible. The above table 2 gives the key sizes in bit that are said to be equal in terms of security.

B. BREAKING RSA AND ECC

The level of effort for factoring integers and computing elliptic curve discrete logarithms is measured in a unit called MIPS year. The term MIPS year denotes the computational power of a MIPS computer utilized for one year; a million-instruction-per-second processor running for one year, which is about $3x10^{13}$ instructions executed [9]. It is worthy to note that a software attack on ECC appears to be relatively more difficult than that of software attack on RSA.

Table 3: Software Attack on RSA and ECC

RSA	ECC	MIPS years to attack
1024	160	10^{12}
2048	224	10^{24}
3072	256	10^{28}
4096	280	10^{31}
7680	384	10^{47}
21000	600	10^{81}

The above figure in table 3 shows the level of effort required for various values of **n** in bits to factor with current version of the GNFS and to compute a single elliptic curve discrete logarithm using the Pollard-rho method.

C. MANAGEMENT FOR THE FUTURE

According to Moore's law, the computing power increases exponentially. So, cryptographic key sizes have to be increased considerably. This makes it unlikely that today's 1024-bit RSA keys will still be considered secure 30 years from now. Taking the Moore's law into consideration and barring any unforeseen developments, RSA key sizes will increase at a faster rate than those of ECC.

As key sizes increase, so do the sizes of signatures and public keys, and so does the time required to perform cryptographic operations on a particular computing platform. This rate of increase will be considerably faster for RSA than it is for ECC. Clearly, RSA cannot satisfy this requirement, and we are forced to consider ECC as an alternative.

Table 4: Encryption and decryption operating speed

Algorithms	Encryption	Decryption
RSA 1024 ECC 160	03.04 ms 81.16 ms	31.51 ms 62.06 ms
RSA 2048 ECC 224	15.21 ms 111.08 ms	203.65 ms 98.71 ms
RSA 3072 ECC 256	16.86 ms 131.11 ms	703.21 ms 115.43 ms
RSA 4096 ECC 280	18.51 ms 145.00 ms	1594.01ms 195.08 ms
RSA 7680 ECC 384	31.96 ms 180.21 ms	10093.05 ms 244.80 ms

D. ENCRYPTION AND DECRYPTION SPEED

In this part, we compare the RSA and ECC algorithms in terms of encryption and decryption operating speeds for key sizes that are said to be equal in terms of security as stated in table 2 above. The message size was 21 bytes and encrypted output was 152 bytes in case of RSA and 203 bytes in case of ECC. The result of the encryption and decryption operating speed on Intel 1.6GHz system with 1GB of memory in Java SE7 under Windows XP is given in table 4.

From the table 4, we conclude that the encryption process in RSA is optimal even for large key sizes such as 7680 bits. However, for decryption the time taken raises considerably. Both the encryption and decryption speeds of the ECC are optimal even for large key sizes. RSA with 21000 bit key size may not be practical to implement; thus forcing us to use ECC. So we conclude that the use of ECC will offer significant benefits over RSA when more security needs increase as operating speed of RSA with large key size increases exponentially.

XIII. EASIER COMPUTATION

Another factor which distinguishes elliptic curve cryptosystem from RSA is the easier computations required for producing the cryptographic parameters. Therefore, elliptic curve cryptosystem better fits for implementations on devices with reduced system resources such as mobile phones.

XIV. ADVANTAGES AND LIMITATIONS OF SMS-BASED E-BANKING

A. ADVANTAGES:

(i) SMS-based E-banking can be done from any handset as all handsets support SMS.

(ii) SMS-based E-banking saves a lot of customers' time as they need not go to banks for enjoying transactions.

(iii) Relatively, SMS-based E-banking reduces costs as the cost of sending an SMS is very cheap.

(iv) SMS-based E-banking makes a lot of conveniences to the customers as they can perform transaction anywhere and anytime.

B. LIMITATIONS:

(i) An SMS message may consist of a maximum of 160 characters. This is one of the limitations in SMS Banking.

(ii) SMS technology is a store-and-forward based system. So, it does not guarantee delivery of messages.

(iii) Some locations may not have network coverage resulting in the unavailability of SMS-based E-banking.

(iv) Most SIM cards have limited space (about 128KB) and hence low function.

XV. CONCLUSION

This paper discusses SMS-based E-banking Services by means of Elliptic Curve Cryptographic technique. Elliptic curves are believed to provide good security with smaller key sizes. Smaller key sizes may result in faster execution timings for the schemes, which is beneficial to systems where real time performance is an important factor. We also gave estimates of key sizes providing equivalent levels of security for RSA and ECC systems.

REFERENCES

[1] Ranbir Soram, Mobile SMS Banking Security Using Elliptic Curve Cryptosystem, International Journal of Computer Science and Network Security, Vol 9, No. 6, 2009.

[2] Ian Blake, Gadiel Seroussi, Higel Smart, Elliptic Curves in Cryptography, Cambridge University Press, 1999.

[3] Joseph H. Silverman, John Tate, Rational Points on Elliptic Curves, Springer, 1992.

[4] Lawrence C. Washington, Elliptic Curves, Number Theory and Cryptography, CRC Press, 2008.
[5] Henri Cohen, Gerhard Frey, Handbook of Elliptic and Hyperelliptic Curve Cryptography, CRC Press, 2006.
[6] Atul Kahate, Cryptography and Network Security, 2E, Tata McGraw, 2011.
[7] Bhattacharya, Jain, Nagpaul, Basic Abstract Algebra, Cambridge University Press, 2002.
[8] Bruice Schneier, Applied Cryptography, Wiley India, 2007.
[9] William Stallings, Cryptography & Network Security, PHI, 2006
[10] Joseph H. Silverman, The Arithmetic of Elliptic Curves, Springer, 1986.
[11] Ian Blake, Gadiel Seroussi, Higel Smart, Advances in Elliptic Curve Cryptography, Cambridge University Press, 2005
[12] Thomas Koshy, Elementary Number Theory with Applications, Academic Press, 2009.
[13] Erdinc Ozturk, "Low Power Elliptic Curve Cryptography" M.Sc thesis, Worcester Polytechnic Institute, April 2004
[14] Menezes, Okamoto, Vanstone, "Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field, IEEE Transaction on Information Theory, vol. 39, 1993.
[15] GSM from the Wikipedia website. [Online]. Available: http://en.wikipedia.org/
[16] J. J. Shen, C. W. Lin and M. S. Hwang, "A modified remote user authentication scheme using smart cards," IEEE Trans. Consumer Electronic, vol. 49, no. 2, pp. 414-416, May 2003.
[17] R. Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley, 2001.
[18] Neal Koblitz, Alfred J. Menezes, "A survey of public-key cryptosystems,", Aug 7. 2004.
[19] Rotman, Galois Theory, Springer International Edition, 2010.
[20] R.L.Rivest, A.Shamir & L.M.Adleman, " A method for obtaining Digital Signature and Public Key Cryptosystems", ACM, 1978.
[21] Kristin Lauter, "The Advantages of Elliptic Curve Cryptography for Wireless Security", Microsoft Corporation.
[22] W. Ford and M. Baum. Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption. Prentice Hall, 2nd edition, 2000.
[23] ANSI X9.62, "Public key cryptography for the financial services industry – the elliptic curve digital signature algorithm (ECDSA)", 1999.
[24] Rotman, Galois Theory, Springer International Edition, 2010

Ranbir Soram works at Manipur Institute of Technology,Takyelpat, Imphal, India. His field of interest includes Network Security, NLP, Neural Network, Genetic Algorithm, and Fuzzy Logic etc.

Memeta Khomdram works at National Institute of Electronics and Information Technology (Formerly DOEACC Centre), Akampat, Imphal.

Sonamani Takhellambam works at Manipur Institute of Technology,Takyelpat, Imphal, India. His field of interest includes Network Security, Wireless Sensor Network etc.