 CMC, vol.57, no.2, pp.341-352, 2018

Design of ECC based Secured Cloud Storage Mechanism for Transaction Rich Applications

V. Gopinath[1, *] and R. S. Bhuvaneswaran[2]

Abstract: Cloud computing is the highly demanded technology nowadays. Due to the service oriented architecture, seamless accessibility and other advantages of this advent technology, many transaction rich applications are making use of it. At the same time, it is vulnerable to hacks and threats. Hence securing this environment is of at most important and many research works are being reported focusing on it. This paper proposes a safe storage mechanism using Elliptic curve cryptography (ECC) for the Transaction Rich Applications (TRA). With ECC based security scheme, the security level of the protected system will be increased and it is more suitable to secure the delivered data in the portable devices. The proposed scheme shields the aligning of different kind of data elements to each provider using an ECC algorithm. Analysis, comparison and simulation prove that the proposed system is more effective and secure for the Transaction rich applications in Cloud.

Keywords: ECC, SSL VPN, cloud computing, banking, security, transaction rich applications.

1 Introduction

Secure Cloud Storage Mechanism for banking is an ongoing research topic that is yet to become omnipresent. This is due to numerous issues with banking storage mechanism which require immediate attention, such as rewards, precautions, confidentiality, perspective, usability, data management, unpredictable growth in volume of transactions and price reduction. Cloud computing provides services to the banking sector as Software-as-Service (SaaS), Platform-as-a-Service (PaaS), Infrastructure-as-Service (IaaS). The SaaS includes Customer Relationship Management (CRM), accounting, invoicing and ERP. PaaS is a perfect platform for applications. It reduces the cost of IT & decreases the number of hardware devices & software applications and cut down the hosting environment. IaaS is a service which allows a business to procure those assets as a completely redistributed service [Agre (2015); Bejju (2014)].

This paper proposes a safe storage mechanism for transaction rich applications, especially for banking, using Elliptic Curve Cryptography, a public key cryptosystem. Cloud service provider gives a base security but it is not sufficient to handle the financial data in the

[1] Sathyabama University, Department of Sciences and Humanities, Chennai, 600119, India.

[2] Anna University, Department of Computer Science & Engineering, Chennai, 600025, India.

[*] Corresponding Author: V. Gopinath. E-mail: vguru007@gmail.com.

cloud environment. Generally banking and customers need more safeguard for the money transactions like payroll, CRM, accounting, invoicing and ERP etc. Banks cannot afford the risk of the security breach. In the existing system of cloud computing, TRA has only a single layer of security. So we are in need to enhance the security mechanism to provide high level security with cost saving, high performance and bandwidth.

Cloud storage mechanism for TRA (banking) customers can get lots of benefits [Rani and Gangal (2012)]. A few of these are below [Patani, Kadam and Jain (2014)]:

- Utilization of Time: Customers can use 24*7 h, so it is very convenient and a great choice to operate financial services for many mobile phone owners in urban areas cause everywhere is cloud.
- Increase Adaptability: It helps banks to enjoy the promotion of adaptability ratios and operating leverage. Secure cloud based storage mechanism for TRA and business operations can be ample extra effectively aligned; the cloud gives to banking a golden opportunity to decline complexity.
- Decrease Invest Amount: The banks are not ready to invest large amount to purchase software, hardware and related work force for the usage of cloud computing. Bank customers can update their account information using all mobile devices from anywhere [Brown (2014)]. Pay-on-demand model means they invest only for those software (s/w) and hardware (h/w) that they need.
- Security Comparison: ECC based secure cloud storage mechanism for TRA is more sheltered than online and internet banking [Alemu and Omer (2014)]. Accessing our bank's website or using our bank's mobile applications to access our account is highly protected than conducting traditional online banking on computer or laptop.

The designed concept provides the extra layer of security with ECC. When the banking customer connects with Cloud, It supports by connecting the P2P network with additional second layer of security SSL and ECC, established cloud environment and the banking application utilizes the same set of ECC digital keys. ECC ANSI X9.62 [Alemu and Omer (2014)] is utilized for the Cloud based TRA establishment; this security system will provide advanced threats to defeat known and unknown threats.

This paper is organized into five sections. First section enlightens the introduction of secure cloud TRA and about the ECC. Second section enlighten that about the related works like mobile banking and how it is related with cloud computing. Third section proposes architecture of Secured Cloud Storage Mechanism for Transaction Rich Applications. Fourth section explains analysis and experiment results Fifth section concludes with the forthcoming work.

2 Related works

A Mobile bank transaction is a concept in which Cloud Computing [Sriram (2010)] is used for the communication of consumers with the bank through appliances such as mobile. The framework is focused in providing security to implement evolving product and increase the level of protection without disrupting banking Operations [Alemu and Omer (2014)]. Therefore users are convenient in using smart phones and other devices integrated with mobile banking, which helps in online banking transactions like fund

transfer and knowing the account balance and locating an ATM (Automated Teller Machine) [Alemu and Omer (2014)]. Some schemes are proposed to strengthen the protection of Cloud Mobile Banking [Filiol and Irolla (2015)] data in centers, but these schemes substantially concentrate on designing some algorithms to keep data confidential.

In cloud mobile banking most of banks are using RSA algorithms to provide security, when the Handshake is established and RSA algorithm is inbuilt with SSL encryption/decryption keys [Jadhao and Kumbhalkar (2016)]. Whenever the users connect Private cloud via VPN to access the financial data, here the SSL uses RSA 128-bit security keys. The ad-vantage of using RSA keys, it is well established. The disadvantage is the bigger key size, so CPU consumption and storage space is high. In the cloud environment [Abbas (2015)] the essential of using crypto system is necessary to use public-key operations for all transactions, the cost of encryption and hashing is fully depends on the amount of data transferred. The Elliptic Curve Cryptography [Mahto, Khan and Yadav (2016)] is required as shorter key size, less CPU consumption and storage space. In this paper cloud mobile banking using ECC algorithm [Alowolodu, Alese and Ogundele (2013)] for secure and more efficient encryption algorithm than RSA as it uses minor key sizes for equal level of security as compared with RSA. Example 256-bit ECC encryption key and 3072-bit RSA encryption key provides same differentiate security. The purpose of the work is to use the ECC algorithm for protective connection of banking system on the cloud.

3 Design of secure storage for transaction with ECC

Architecture of Secure Cloud Transaction rich application is explained in Fig. 1. Moveable equipments and Remote system are connected to the mobile operators and Wireless Access Point. These are connected to either entry point or satellite via base station. So the Mobile users request and response are delivered to a cloud banking through existing SSL VPN with extra protection of ECC. The projected security system is linked with Internet Service Providers (ISP) and finally the financial sectors being connected to ISP.

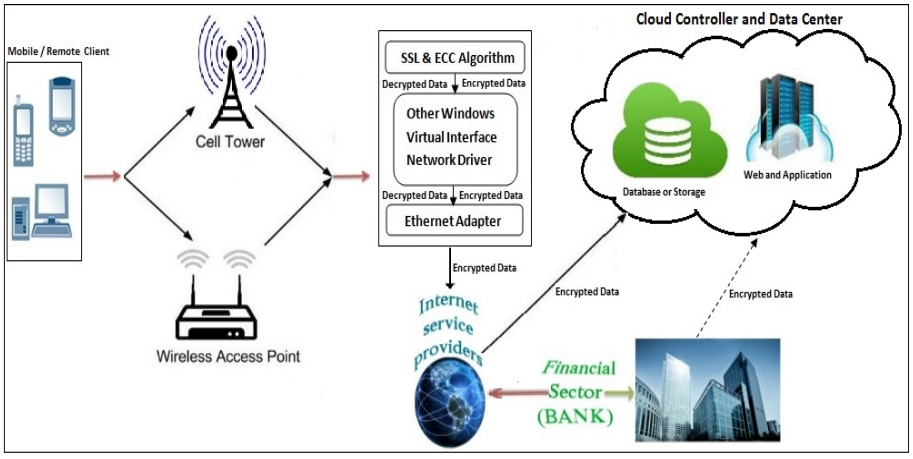

Figure 1: Architecture of secure TRA

Central processors which are linked to mobile network receive the mobile users request as well as send details such as user id and user location to banking Database storage. The projected security theory enhances the level of protection that currently supports the Cloud Mobile banking and facilitates the realization of P2P Network.

Mobile users can take the banking facility 24/7 from cloud computing as corresponding bank details are present on data centers of cloud computing. The proposed system provides the extra layer of security for storing the sensitive data in trusted private cloud via SSL VPN with ECC. The major objective of the proposed security framework is as follows:

- To assure information security and privacy protection of entire mobile/remote users activities in the cloud. ECC based SSL VPN is being utilized in to the system.
- ECC encryption helps us to provide less bandwidth, computing power and memory for creating customized security for Data packets.

Implementing an extra layer of security system over SSL VPN with ECC to safeguard the integrity and confidentiality of mobile user's private data to create a key that is difficult to decrypt and thereby making it highly difficult to hack the network, so the user is more protected to connect to the banking system.

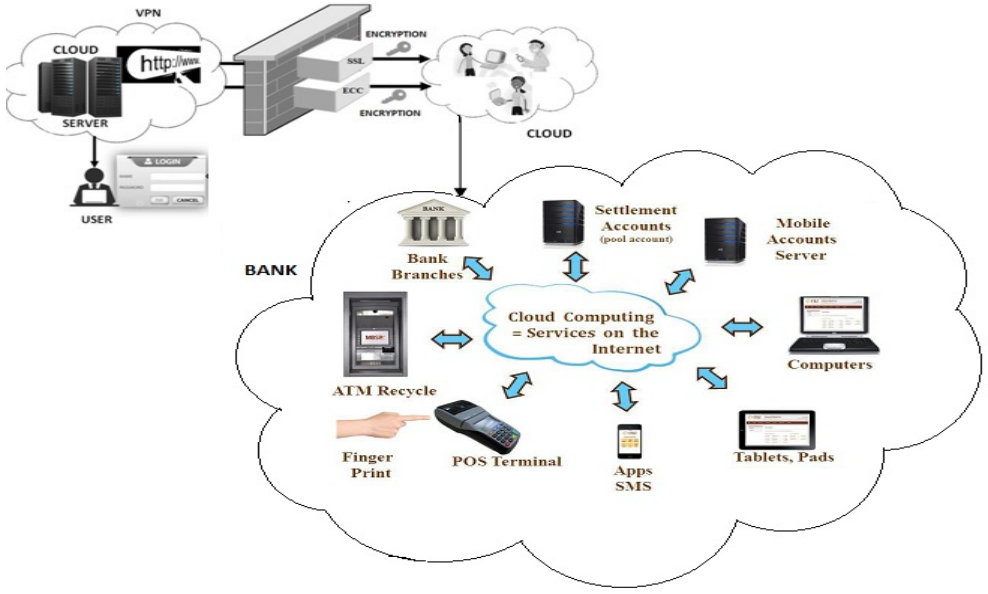

Figure 2: Cloud TRA with 2 level securities

The above Fig. 2 depicts that the user login into the private cloud VPN using 2 levels of security like ECC and SSL, after the handshake same set of Encryption/Decryption keys has been utilized. ECC is endowed with an extra protection along with the existing SSL VPN, which is applied on a Cloud. It helps in linking the cloud mobile banking network through an ECC.

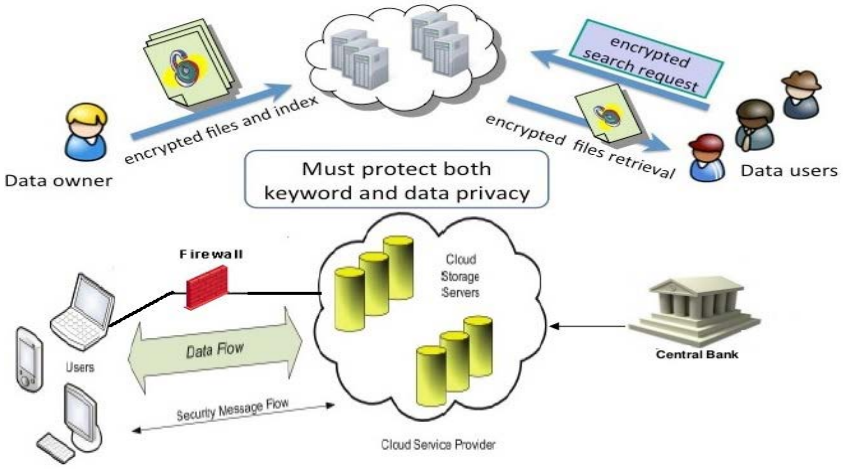

Figure 3: Cloud TRA with security level

The above Fig. 3 depicts cloud mobile banking architecture with security level. The three different identifiable network entities are as follows: User, Cloud Service Provider like Amazon Web Services [Amazon Web Services (2010)] and Central bank [Bejju (2014)]. Data user's posses the way in to update/edit general data like insertion, deletion, modification, appending, reordering etc. which needs additional layer of security to protect the data. The exchange of data from user to owner and in reverse is via Firewall and Cloud server provider. ECC plays an essential role in providing a secured message flow.

3.1 Cloud storage mechanism for TRA

The cloud storage mechanisms are designed exclusively for cloud-based supplying. Like how the physical server can release virtual server images, similarly device's instances can be virtualized. They are commonly able to provide fixed-increment capacity allocation in support of the pay-per-use mechanism. The stored data can be exposed for remote access via cloud storage devices. It provides general logical elements of data storage, such as:

- Group of data that are stored in the folders are called as files
- The smallest unit of data which can be individually accessible and is the lowest level of storage and the closest to the hardware is called a Block.
- The collections of data there are organized into a table-based, delimited, or record format is called as dataset.
- Objects-Web-based resources are the organized data and its associated metadata.

Lack of control and risk for the customer are mainly caused by distributed data storage as there is high risk for data loss, it is very essential to recover the data when some problem occurs and creates failure. But customers must be ensuring that their data will remain available even after such an event. Providing data security in cloud environment is very much essential. The proposed design will provide the secure cloud storage mechanism for TRA.

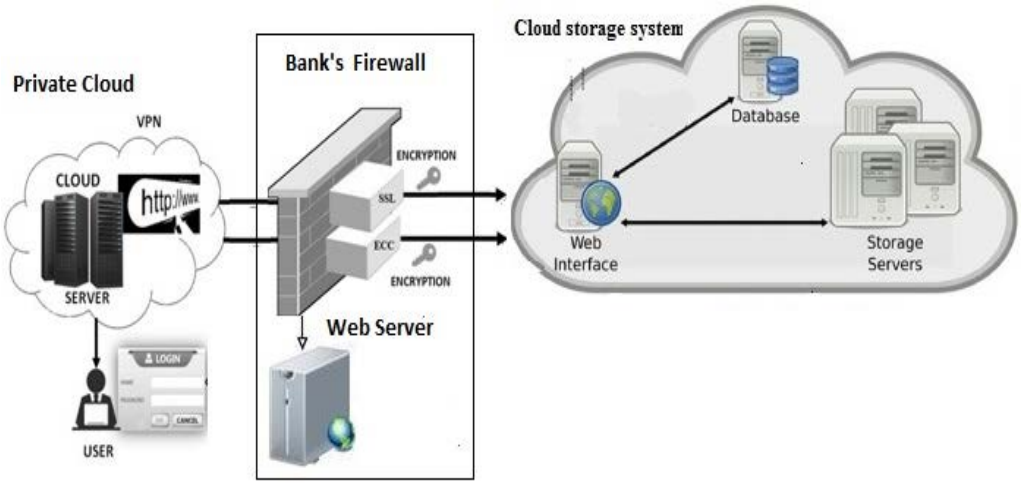

Figure 4: Cloud storage mechanism for TRA

The above Fig. 4 shows the data request of the user via internet to access the private cloud data servers and applications to utilize the Bank's web server and firewall. The design is experimented with customized java coding and inbuilt ECC algorithm into the web server, logs are maintained in the DB tables as well as web server and user can access the web server system with help of SOAP UI, user can hits directly to the bank's web server and firewall and data packets are send as an encrypted files for both request and response from the cloud storage system.

Many banks are already applying cloud computing for non-core and non-critical activity, such as development, testing etc. Quite a few Small finance banks either have transferred or in the process of transferring entire core services to the cloud. It makes the banks services more convenient, accessible, easier to use, and personalized to the individual's needs and their lifestyle, users have no knowledge about the hosted data.

3.2 Proposed algorithm of ECC

In the proposed system, the required java coding and its manipulation is done with inbuilt ECC algorithm.ECC is an approach to public-key cryptography, it is smaller, faster with well-organized cryptographic keys. The time-honored method creates the huge prime numbers but elliptic curve equation generates keys from ECC effectively. Research says, normally other system requires 7680 bit to provide security were as ECC requires only 384-bit key to acquire the secuirty. ECC is vastly used in mobile application as it gives hand to lay foundation equivalent security with minimized computing capacity and battery resource usage.

ECC works at elliptic curve defined over finite field $\mathbf{F_{cp}}$. General Equation of elliptic curve is:

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

Here a, b, x and y are real numbers and elliptic curve changes with various choices of a and b, which is defined over the finite field Fcp, Where cp is prime number, x and y are the elements of E ($\mathbf{F_{cp}}$).Point addition is addition of two points. Suppose point $\mathbf{CP_3}$

represented by $CP_3 = CP_1 + CP_2$, where CP_1 and CP_2 are points on the elliptic curve can be found by drawing the line between CP_1 and CP_2 [Agrawal and Gera (2014)]. The point where the line intersects the elliptic curve is taken and reflected across the curve's horizontal line of symmetry, which much of the time is the x-axis. The resultant Point is the sum of CP_1 and CP_2. Point addition can also be defined by the following equations.

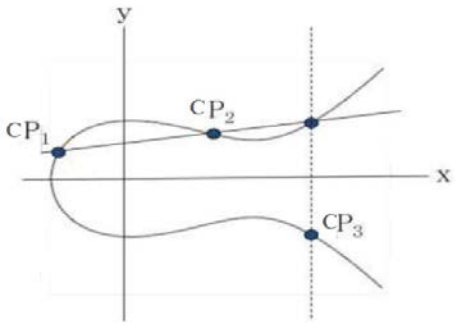

Figure 5: Elliptical curve

As shown in Fig. 5, let $CP_1 = (x_1, y_1)$, $CP_2 = (x_2, y_2)$, $CP_3 = (x_3, y_3)$ and CP_1 not equals CP_2.

$$m = \frac{y_2 - y_1}{x_2 - x_1} \tag{2}$$

To find the insertion with E. we get

$$(\mathrm{m}(x - x_1) + y_1)^2 = x^3 + Wx + Z \tag{3}$$

So,

$$x_3 = m^2 - x_2 - x_1$$

$$y_3 = \mathrm{m}(x_2 - x_1) - y_1 \tag{4}$$

Both clients consents to some publicly aware of information items.

- The elliptic curve mathematical statement
- Estimation of W and Z
- Prime Numbers P
- The elliptical curve figure gathered from the elliptic curve equation and W base point, Z, taken from the elliptic gathering.

3.2.1 Key generation

W chooses a whole number dW. This is W's private key.

- W then produce a public key PW=dW×Z
- Z correspondingly chooses a private key dZ and process an public key PZ=dZ×Z
- A produces a security key K=dW×PZ
- Z produces the security key K=dZ×PW
- B produces the security key K=dZ×PW.

3.2.2 Signature generation

For marking a message m by W, utilizing W‘s private key dW

- Compute e=HASH (m), where HASH means cryptographic hash function, such as SHA-1
- Select u arbitrary whole number k from [1, n-1]
- Compute r=x1(mod n), where(x1, y1) =k×Z. If r=0, go to Step 2
- Computes=k-1(e+dWr) (mod n). If s=0, go to Step 2
- The signature is the couple of (r, s)
- Send signature (r, s) to Z client

3.2.3 Encryption algorithm

Suppose W wants to send to Z an encrypted message.

- W takes plaintext message M, and encodes it onto a point, PM, from the elliptic gathering
- W picks another arbitrary whole number, k from the interval [1, p-1]
- The cipher text is a couple of points
- PT=[(kZ), (PM+kPZ)]
- Send cipher text PT to client Z.

3.2.4 Decryption algorithm

Client Z will take the following steps to decrypt cipher text PT.

- V computes the result of the principal point from PT and private key, dZ=dZ×kZ
- V then takes this item and subtracts it from the second point from PT
- (PM+kPZ)-[dZ (kZ)]=PM+k(dZZ)-dZ(kZ)=PM
- Z cloud then deciphers PM to get the message, M.

3.2.5 Signature verification

For Z to authenticate W‘s signature, Z must have W‘s public key PW

- Confirm that r and s are whole numbers in [1, n-1]. If not, the signature is invalid
- Evaluate e=HASH (m), where HASH is the same function used in the signature generation
- Evaluate w=s-1(mod n)
- Evaluate u1=ew(mod n) and u2=rw(mod n)
- Evaluate (x1, y1)=u1B+u2PW
- The signature is valid if x1=r (mod n), invalid otherwise.

Here n is the smallest positive integer, and is equal to the ratio #E ($\mathbf{F}_{cp}$)/n, where # E ($\mathbf{F}_{cp}$) is the curve order. The most important security in ECC is the parameter n which is the order of the point. This determines the strength and level of the security of the system. The model will be extended with the introduction of Secured Socket Layer (SSL) to further secure the data sharing tunnels where each user’s has two keys a “public” key and a “private” key. Anything encrypted with the user’s public key can only be decrypted with the private key and vice versa.

4 Result and performance analysis

This section presents the result and performance analysis for the proposed Secure Cloud storage mechanism for transaction rich applications, the design is experimented with customized java coding and inbuilt ECC algorithm which is deployed in Cloud environment. We have used SOAP UI tool for performance analysis, it is deployed into the cloud web server. With the help of this, we can check the performance and load balance testing. Single usage of customized SOAP UI can trigger 1000 users at a time in the Cloud storage server. Logs are stored in the database table as well as in the Server and client system. Based on the load balance testing and performance tracking the below Tab.1 shows about the various comparison of proposed and existing system in the Cloud storage mechanism for TRA.

Table 1: Analysis report and comparison between existing framework and proposed framework

S No.	Analysis report	Proposed system	Existing system
1	Implementation time	Short	In general, Significantly longer
2	Upfront Investment	Low investment	High investment
3	Additional Hardware/IT costs	Not required	Yes, Required
4	All-Time costs	Predictable cost	Unpredictable cost(but maybe lower)
5	Degree of customizations	Less customizable in general	Greater ability to customize
6	Control of data security standards	Customer can control the data	banks can control the data
7	Confidentiality	Symmetric key ,	File encryption not provided
8	Authentication	Password-based advance level security provided.	Only base level security provided.
9	Access Control	Encryption of security area information	Exposure to the normal area
10	Log Security in Virtual Machine, Data Packets	Provided and It is Secured	Not provided

Cloud TRA has the SSL protocol inbuilt with RSA, in this proposed system, RSA is replaced by ECC. ECC is an open key cryptography technique providing 384-bit Security key [Mahto, Khan and Yadav (2016)], the effectual outlay of public key procedure is constant by the frequency of session reuse which reduces the requirement for open key operations in some transactions. The charge of encryption and muddling depends on the volume of information conveyed. The reason being that ECC provides greater efficiency in terms of computational overheads, key sizes and bandwidth. MIPS (million instructions per second) are implemented for comparing the speed of different computers. From the Tab. 1, it is clear that ECC affords the same security as RSA while using significantly smaller key sizes.

Table 2: Key size strength for ECC

Time to Break (MIPS-Years)	Proposed ECC Algorithm Key Size	Existing RSA Algorithm Key Size	Key Size Ratio
1.00E+12	160	1024	1:07
1.00E+14	224	2048	1:10
1.00E+28	256	3072	1:12
1.00E+47	384	7680	1:20
1.00E+66	512	15360	1:30

The above Tab. 2 MIPS 1e+12 means one times ten to the power 12, ECC and RSA key size strength ratio measure in the above table for example RSA system requires 7680 bit to provide security were as ECC requires only 384-bit key to acquire the security and it key ratio stands with 1:20. So ECC provides greater efficiency in terms of key size and bandwidth, it means higher speed and lower power consumption.

Table 3: Measure of proposed ECC *vs.* existing RSA public keys

Algorithm	Key size	Key Generation Time (ms)	Required Memory Size (bytes)	Encryption/Decryption Time (ms)
Proposed ECC Algorithm	160	108	125	16
	224	121	140	15
Existing RSA Algorithm	1024	2609	313	388
	2048	18399	621	1867

The public key operation for ECC-160 is only 3.69 milliseconds, it is 50% comparatively lesser than RSA-1024 and other keys. The flowchart given below explains that a key generation time and required memory size for both ECC key and 1024-bit RSA key, 160-bit ECC is much better than RSA [Abbas (2015)]. The protection measures for both 160-bit ECC key and 1024-bit RSA key is similar. Hence breaking a 160-bit key, would be a hundred million times harder than breaking the 1024-bit key.

The OpenSSL speed plan can be utilized to evaluate RSA decryption and ECC function for various key sizes. Outcome of the projected system is shown below.

Fig. 6 explains the theoretical and practical security analysis of ECC and RSA on the basis encryption and decryption times on the sample data of 160 bit, 224 bit, 256 bit, 384 bit and 512 bit and justifies that ECC is overall more efficient and secure than RSA.

During the ECC key generation [Khabbazian (2004)], time taken to generate RSA key and ECC key does not differ much, where as there is a huge difference in the size of the keys (ECC-384 and RSA-7680). Also we understand that during the signature generation process ECC surpass RSA. Conversely RSA beats ECC in performance during the verification process 5.

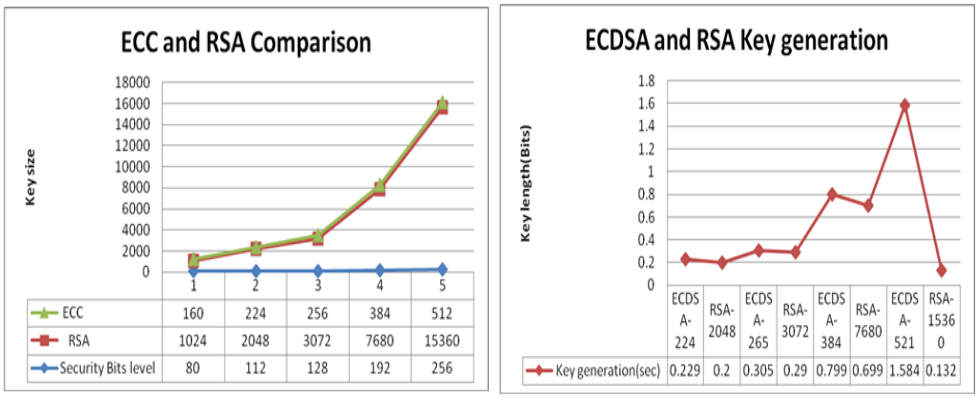

Figure 6: ECC *vs.* RSA key comparison and generation

5 Conclusions

Continuously increasing the advancement of cloud within the mobile's banking technology provides many features for the mobile users and bank customers. In this article, the use of Secure Cloud Transaction Rich Application (SCTRA) is explained. The pay roll, bills, fund transfer and payments can be easily transacted by mobile banking. In the existing private cloud the stage of safeguard is intensified by the proposed security design. It shelters data transmission over the consolidated network direction by exploiting the regular P2P network over the cloud TRA. All the outcome displays that the suggested scheme is effective and feasible to defend the Cloud Portable bank transaction. Cloud computing has some deployment models which can be used in banking sector. The architecture of cloud mobile banking explains about how mobiles and banks are connected with cloud and how they work to meet the requirements of different services with secured communication. Mobile banking using cloud computing has more benefits for the banking customers. Along with the benefits, some disputes are also present to fight and some problems also exist in cloud mobile banking.

References

Abbas, S. A. (2015): Improving data storage security in cloud computing using elliptic curve cryptography. *IOSR Journal of Computer Engineering*, vol. 17, no. 4, pp. 48-53.

Abimbola, A. (2013): Advantages and disadvantages of mobile banking. http://mauconline.net/2013/03/07/advantages-anddisadvantages- of-mobile-banking.

Agrawal, K.; Gera, A. (2014): Elliptic curve cryptography with hill cipher generation for secure text cryptosystem. *International Journal of Computer Applications*, vol. 106, no. 1.

Agre, C. (2015): Implementation of a cloud in banking sector. *International Journal of Computer Science and Information Technology*, vol. 3, no. 2, pp. 1168-1174.

Alemu, M.; Omer, A. M. (2014): Cloud computing security framework for banking industry. *HiLCoE Journal of Computer Science and Technology*, vol. 2, no. 1, pp. 79-85.

Alowolodu, O. D.; Alese, B. K.; Ogundele, O. S. (2013): Elliptic curve cryptography for securing cloud computing applications. *International Journal of Computer Applications*, vol. 66, no. 23.

Amazon Web Services: Elliptic curve cryptography and forward secrecy support in AWS IoT. https://aws.amazon.com/blogs/iot/elliptic-curve-cryptography-and-forward-secrecy-support-in-aws-iot-3/.

Bejju, A. (2014): Cloud computing for banking and investment services. *Advances in Economics and Business Management*, vol. 1, no. 2, pp. 34-40.

Brown, S. (2014): Six reasons why cloud computing will transform the way banks serve clients-and the five hurdles to overcome, banking technology. http://www.bankingtech.com/236322/six-reasons-why-cloudcomputing-will-transform-the-way-banks-serve-clients-and-the-five hurdles- to-overcome/.

Chintawar, N. N.; Gajares, S. J.; Fatak, S. V. (2016): Cloud data security enhancing using elliptical curve cryptography. *Digital Communications and Networking Commons*, pp. 30-35.

Jadhao, A. S.; Kumbhalkar, S. B. (2016): Technical review on secure banking using RSA and AES encryptor methodologies. *IOSR Journal of Electronics and Communication Engineering*, vol. 11, no. 1, pp. 1-4.

Khabbazian, M. (2004): Software elliptic curve cryptography. http://hdl.handle.net/1828/515.

Mahto, D.; Khan, D. A.; Yadav, D. K. (2016): Security analysis of elliptic curve cryptography and RSA. *Proceedings of the World Congress on Engineering*, vol. 1.

Patani, S.; Kadam, S.; Jain, P. V. (2014): Cloud computing in banking sector: A survey. *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 2, pp. 5640-5643.

Rani, S.; Gangal, A. (2012): Security issues of banking adopting the application of Cloud computing. *International Journal of Information Technology and Knowledge Management*, vol. 5, no. 2, pp. 243-246.

Sriram, S. (2010): Cloud computing in banking. *Capgemini Financial Services*, pp. 1-12.