Performance Analysis of RSA and Elliptic Curve Cryptography

Dindayal Mahto and Dilip Kumar Yadav
(Corresponding author: Dindayal Mahto)

Department of Computer Applications, National Institute of Technology Jamshedpur
Jamshedpur-14, Jharkhand, India
(Email: dindayal.mahto@gmail.com)

(Received Mar. 27, 2017; revised and accepted Sept. 21, 2017 & Nov. 5, 2017)

Abstract

This paper presents a performance study and analysis of two popular public-key cryptosystems: RSA with its two variants, and ECC (Elliptic Curve Cryptography). RSA is considered as the first generation public-key cryptography, which is very popular since its inception while ECC is gaining its popularity recently. Besides studying and analyzing the paper also suggests the supremacy among these cryptosystems based on the experimentation. The paper shows the result of the experimentation performed using these cryptosystems with the different modulus/key sizes recommended by the NIST. The modulus/key sizes are used such as 1024/2048/3072-bit for RSA and 160/224/256-bit for ECC. After experimentation and execution of these cryptosystems, the paper concludes that an ECC-based cryptosystem is better than an RSA or its variants-based cryptosystem, and an ECC based cryptosystem best suits for memory-constrained devices, as an ECC-based cryptosystem requires fewer resources than an RSA-based cryptosystem.

Keywords: Decryption; Elliptic Curve Cryptography; Encryption; Public-Key Cryptography; RSA

1 Introduction

Asymmetric key cryptography or public-key cryptography (PKC) uses two keys mainly a private key and a public key; the private key is used for decryption or signature generation while the public key is used for encryption or signature verification. The PKC gains its popularity by developing two pioneering concepts, the firstly, solving key distribution problem of symmetric key cryptography and, then secondly, providing a digital signature scheme [12, 18, 23]. This type of cryptography is mostly used by all leading social and commercial websites for exchanging keys (*i.e.*, small data) in a secure way, and achieving authenticity, integrity, and non-repudiation services. For example, ECDHE_RSA protocols (Elliptic Curve Diffie-Hellman Key Exchange with RSA) are being used by www.amazon.in, and www.linkedin.com, and ECDHE_ECDSA protocols (ECDHE with Elliptic Curve Digital Signature Algorithm) are being used by www.facebook.com, and www.mail.google.com.

RSA [38] is considered as the defacto standard for the public-key cryptography, while ECC [20, 33, 46] is considered as an alternative to RSA. The security of RSA cryptosystem is based on the Integer Factorization Problem (IFP) and the security of ECC is based on the Elliptic Curve Discrete Logarithm Problem (ECDLP). The main attraction of ECC over RSA is that the best-known algorithm for solving the ECDLP takes full exponential time while to solve the IFP of RSA takes sub-exponential time. The fastest algorithm is known as Pollard's rho algorithm for solving the ECDLP takes full exponential time, which has an expected running time $\sqrt{\pi n}/2$. As on 2003, the largest ECDLP instance solved with Pollard's rho algorithm for an elliptic curve over a 109-bit prime field. The best-known generic integer factoring method is Pollard's general number field sieve (NFS). The heuristic expected run-time needed for the NFS to find a factor of the composite number n is L[n] = [1/3, 1.923]. The largest integer factored using the NFS takes sub-exponential time, is the RSA200, a 200-digit number (665-bit) which was factored in May 2005 [16]. This means that, for the same level of security, significantly smaller parameters can be used in ECC than RSA. For example, to achieve 112-bit of security level, an RSA based cryptosystem needs a key of a size of 2048-bit, while an ECC based cryptosystem needs a key of a size of 224-bit [2] as shown in Table 1 and Figure 1. This paper demonstrates the usage of the algorithms of RSA and ECC between two communicating parties (*i.e.*, Alice and Bob).

This paper is organized as follows. In Section 2, the related works and literature reviews are described. In Section 3, RSA and its variants algorithms are described. In Section 4, ECC algorithm is described. In Section 5, different case studies are stated, in Section 6, a performance analysis of RSA and its two variants with ECC are mentioned and in Section 7, the conclusion is stated.

Table 1: Key size (NIST recommended) [2]

Security Bits level	RSA	ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

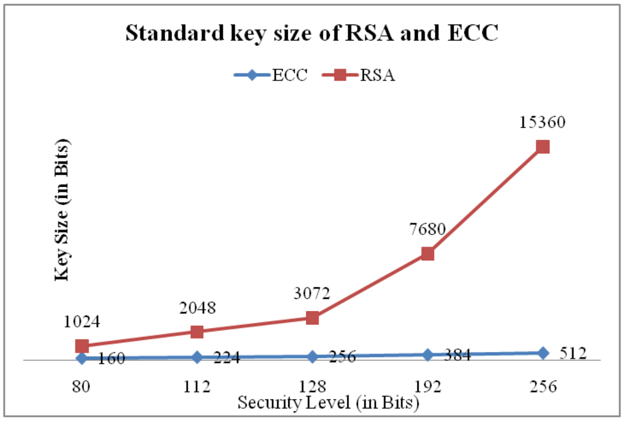


Figure 1: Key size (NIST recommended) [2]

2 Related Works and Literature Reviews

The security/performance analysis of RSA and ECC with different parameters of measurements have been presented by many authors. Gura *et al.* [14] compared point multiplication operation of an elliptic curve over RSA and ECC on two 8-bit processors computer systems and they found on both systems that ECC-160 point multiplication is more efficient than RSA-1024 private-key operation. Bos *et al.* [6] presented an assessment of the risk of the key for RSA and ECC based on key length, and they concluded that till 2014, the use of 1024-bit RSA provides some small risk, while the 160-bit ECC over a prime field may safely be used for a much longer period. Kute *et al.* [21] concluded that RSA is faster but security wise ECC outperforms RSA. Jansma *et al.* [19] compared the usages of digital signatures in RSA and ECC and suggested that RSA may be a good choice for the applications, where verification of a message is required more than a generation of the signature. Alese *et al.* [1] suggested that currently, RSA is stronger than ECC although, however, near future, ECC may outperform RSA. Mahto *et al.* [24–31] demonstrated that ECC outperforms in terms of operational efficiency and security over RSA.

3 RSA and Its Two Popular Variants

Boneh *et al.* [5] presented a survey of four variants of RSA designed to speed up RSA decryption and speed efficiency of these variants using a 1024-bit RSA modulus. They stated that a batch RSA and two multi-factor RSA methods ($n = p^2q$ and n=pqr) are supposed to be fully backward-compatible. They also stated that the rebalanced RSA method provides more speed up with large encryption-exponent 'e'. Chang *et al.* [7] presented a parallel implementation for generating RSA keys using an alternative of the Euclidean Algorithm *i.e.*, Derome's method. The paper claimed the proposed protocol works at low computational cost. Verma *et al.* [47] claimed that modulus and the key generation are achieved using a small order of matrix. In order to generate approximately 840-bit modulus and a private key of RSA, a matrix of four orders is enough. The paper implemented a model in which a small encryption exponent is used to speed up encryption whereas Chinese Remainder Theorem (CRT) is used to speed up decryption time. Ahmad *et al.* [32] proposed a variant of RSA encryption that uses CRT to conceal more than one plaintext in one ciphertext. They proved that their algorithm is safe against several security attacks and proposed some solutions for other security attacks. Santosh *et al.* [41] claimed that they can break the Multi-prime RSA using lattice basis reduction when a user generates 'n' instances with the same modulus. Dong *et al.* [13] proposed to improve threshold secret sharing schemes based RSA with CRT and they claimed that the security channel is not required for their scheme, as each participant chooses his secret shadows by himself as well as the participant can verify the authenticity of secret shadows generated by other participants. Takayasu *et al.* [45] provided enhanced lattice construction for the (δ, β)-SIP and their result shows that if differences of prime factors are small, then the Multi-Prime RSA is vulnerable than the expected.

3.1 RSA (Basic)

RSA (Basic) or RSA [38] is considered as the first real life and practical asymmetric-key cryptosystem. The algorithm (Algorithm 1) for RSA is given below. The security of RSA lies with integer factorization problem.

Here, the key generation is done by each party, once key generation gets over, they can communicate each other securely. In RSA algorithm, for encryption, an exponent e should be chosen such that gcd($\Phi(n)$, e) is equal to 1, and for decryption an exponent, d is generated with the help of finding the inverse of e mod $\Phi(n)$.

In encryption process, the sender has to encrypt the message (*i.e.*, in decimal digit) with the help of the receiver's public key *i.e.*, e and n. In decryption process, the receiver has to decrypt the ciphertext with the help of his own private key *i.e.*, d and n.

```
Algorithm 1 : RSA (also called RSA (Basic))
RSA algorithm exhibits key generation, encryption, and decryption.
Key Generation
1: Select p, and q; where, p and q both are primes, $p \neq q$.
2: Calculate $n = p \times q$.
3: Calculate $\Phi(n) = (p-1) \times (q-1)$.
4: Select encryption exponent e;
   gcd($\Phi(n)$, e) = 1 and (1 <e <$\Phi(n)$).
5: Calculate decryption exponent d;
   $d \equiv e^{-1}(mod\ \Phi(n))$.
6: Public key PU = (e, n).
7: Private key PR = (d, n).
Encryption
1: Plaintext: M < n.
2: Ciphertext: $C = M^e$ mod n.
Decryption
1: Ciphertext: C.
2: Plaintext: M=$C^d$ mod n.
```

3.2 RSA with Chinese Remainder Theorem (CRT)

This method [37] presented a method to break the decryption exponent $i.e., d$ into two parts (d_p, d_q) to decrease the decryption time of RSA, using CRT. Using this technique RSA decryption achieves 4 times faster than RSA (Basic). The algorithm (Algorithm 2) for RSA with CRT is given below.

```
Algorithm 2 : RSA with CRT
RSA with CRT algorithm exhibits RSA with decryption using CRT.
Key Generation
1: Same as RSA (Basic).
Encryption
1: Same as RSA (Basic).
Decryption
1: Calculate $d_p$ = d mod p-1, and $d_q$ = d mod q-1.
2: Calculate $M_p = C^{d_p}$ mod p, and $M_q = C^{d_q}$ mod q.
3: Calculate M from $M_p$, and $M_q$ using CRT.
```

RSA with CRT improves the overall efficiency of RSA.

3.3 Multi-prime RSA

This variant [10] of RSA, further tried to decrease the decryption time with the help of forming modulus 'n' using multiple primes instead of only two primes. It used k primes: p1, p2, . . . , pk. The algorithm (Algorithm 3) for Multi-prime RSA is given below.

```
Algorithm 3 : Multi-Prime RSA
The Multi-prime RSA algorithm exhibits key generation using multiple primes, encryption, and decryption using CRT.
Key Generation
1: Calculate n = $\prod_{i=1}^{k} p_i$, where, k distinct primes p1,
   p2, . . . , pk, each one [n/k]-bit in length. For a 1024-
   bit modulus one can use at most k=3 ($i.e.$, n = pqr).
2: Calculate $\Phi(n) = \prod_{i=1}^{k}(p_i$-1).
3: Select e and d as done with RSA (Basic).
4: Calculate $d_i = d\ mod(p_i - 1), where, 1 \leq i \leq k$.
5: Public key PU = (e, n).
6: Private key PR = (d1, d2, . . . , dk).
Encryption
1: Same as RSA (Basic).
Decryption
1: Calculate $d_p$ = d mod p-1, $d_q$ = d mod q-1, and $d_r$ =
   d mod r-1.
2: Calculate $M_p = C^{d_p}$ mod p, $M_q = C^{d_q}$ mod q, and
   $M_r = C^{r_q}$ mod r.
3: Calculate M from $M_p$, $M_q$, and $M_r$ using CRT.
```

4 Elliptic Curve Cryptography (ECC)

An ECC over a prime field is defined by following general equation in two variables with coefficients.

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

where, a and b are the coefficient of the elliptic curve, and the discriminant, $\Delta = 4a^3 + 27b^2 \neq 0$. The $\Delta \neq 0$ requires to form a group and hence to implement cryptography using elliptic curve.

An ECC is another promising asymmetric key cryptosystem, independently coined by Miller [33] and Koblitz [20] in the late 1980s. For better and stronger security of data, bigger key sizes require, which means more overhead on the computing systems. Nowadays small devices are playing important role in the digital world, however, these devices have less memory as well as they also require security. In this scenario, RSA becomes second thoughts. An ECC based system is most suitable for memory constraint devices such as Palmtop, Smartphone, Smartcards, etc. For an equivalent level of security, an ECC requires comparatively less or smaller parameters for encryption and decryption than RSA cryptosystem. Bhardwaj *et al.* [4] implemented the algorithms for ECC for point doubling, point addition, scalar multiplication. They also measured the performance of the ElGamal encryption and decryption using Elliptic Curve over a Finite Field. Qian *et al.* [36] did a study of an ECC based Radio Frequency Identification (RFID) security protocol and highlighted some features like, an ECC provides realistic security for communication and tag memory data access, it also reduces the key storage requirement and the backend system by storing private key only, the protocol uses XOR, bitwise AND, so forth which further reduces the tag computation, and at the end, the BAN-logic is used to discuss computational performance, security features, and formal proof of the protocol. Basu [3] presented a transformation algorithm that reduces the number of elementary operations, whereas a parallel computation and the concatenation stages reduce the computational cost using elegant parallel implementation. This simulation shows that the speed attains value nearly equal to the order of N, where N is the number of processors. Srinath *et al.* [44] proposed an Undeniable Blind Signature true Scheme (UBSS) based on the features of isogenies between super-singular elliptic curves and proved that their scheme is safe in the presence of a quantum adversary under certain assumptions. Hou *et al.* [17] proposed a robust and efficient remote authentication scheme with the help of an ECC using CAPTCHA technique and provided a formal proof of the scheme using the BAN-logic. Han *et al.* [15] proposed a new authentication scheme to protect user anonymity and insecure against impersonation attack. They compared their scheme with recent schemes and claimed that their scheme can provide stronger security and more efficiency. Naresh *et al.* [34] proposed an ECDLP based dynamic contributory group key agreement protocol for secure group communication over adhoc networks. Liu *et al.* [22] presented that the algebraic structure of bilinear groups loses the advantages of ECC which gains mainly from smaller parameter size and hence they claimed that this structure is not fit for to cryptographic schemes. The algorithm (Algorithm 4) for ECC is given below.

Here, a P_m is an x, y point encoded with the help of a plaintext message, 'm'. This type of different points is used for encryption and decryption in ECC.

This illustration (Algorithm 5) exhibits a data communication security model for an (OTP) One-Time Password (*i.e.*, "32145688") message using an ECC based cryptosystem.

5 Different Case Studies of Implementation of RSA or/and ECC in Software Security, Hardware Security, Wireless LAN Security

5.1 Implementing Software Security

Public-key cryptography provides two important services of information security. They are as follows:

- Secrecy of information: It is provided using encryption and decryption algorithms.
- Authentication of information: It is provided by implementing a digital signature algorithm.

Algorithm 4 : ECC

```
ECC algorithm exhibits key generation, encryption, and decryption.
Global public elements
  1: Chooses an elliptic curve $E_q(a, b)$ with parameters a, b, and q, where q is a prime and $> 3$, or an integer of the form $2^m$.
  2: Selects G(x, y) - a global point on elliptic curve whose order is large value n.
Alice key generation
  1: Selects a private key, $V_A$; where, $V_A < n$.
  2: Calculates the public key, $P_A(x, y)$;
     $P_A(x, y) = V_A \times G(x, y)$.
Bob key generation
  1: Selects a private key, $V_B$; where, $V_B < n$.
  2: Calculates the public key, $P_B(x, y)$;
     $P_B(x, y) = V_B \times G(x, y)$.
Secret key calculation
by Alice
  1: $S_K(x, y) = V_A \times P_B(x, y)$.
Secret key calculation
by Bob
  1: $S_K(x, y) = V_B \times P_A(x, y)$.
Encryption by Alice
using public key of Bob
  1: Alice chooses message $P_m(x, y)$ and a random positive integer 'k' and $1 < k < q$.
  2: Ciphertext, $C_m((x, y), (x, y))$;
     $= ((k \times G(x, y)), (P_m(x, y) + k \times P_B(x, y)))$.
Decryption by Bob
using his own private key
  1: Ciphertext, $C_m((x, y), (x, y))$.
  2: Plaintext, $P_m(x, y)$;
     $= (P_m(x, y) + k \times P_B(x, y))$ - $(k \times V_B \times G(x, y))$
     $= P_m(x, y)$.
     Here, first coordinate of $C_m$ gets multiplied with the private key of the Bob i.e., $V_B$, which in turns becomes similar to Bob's public key. Finally, due to subtraction of resultant coordinate with the second coordinate of the ciphertext $C_m$, all get canceled and only $P_m(x, y)$ gets left.
```

Algorithm 5 : ECC (An illustration of ECC)

```
The key generation, encryption, and decryption of ECC use a 160-bit modulus and key size.
Global public parameters
  1: Consider a prime number q = 52614059007508089314492115406110610700143315430473, a = 0, b = 2,
     G(x) = 1, and G(y) = 2516921478813373080782285662390716255192012539823.
     Based on global public parameters, the elliptic curve equation becomes:
        $y^2 \bmod 52614059007508089314492115406110610700143315430473$
        $= (x^3+2) \bmod 52614059007508089314492115406110610700143315430473.$   (2)
Alice Key Generation
  1: Selects a random private key, $V_A$; The value of $V_A$ is 123456789.
  2: Calculates the public key, $P_A(x,y)$;
     $P_A(x,y) = V_A \times G(x,y) =$ 123456789 * (1, 2516921478813373080782285662390716255192012539823) =
     (41598408633041765117904814957892579387787288121723, 30331027768661217591203589322093387887193929936089).
Bob Key Generation
  1: Selects a random private key, $V_B$; The value of $V_B$ is 987654321.
  2: Calculates the public key, $P_B(x,y)$;
     $P_B(x,y) = V_B \times G(x,y) =$ 987654321 * (1, 2516921478813373080782285662390716255192012539823) =
     (1680504078792863419186290060061493676876263991152, 111209 26205562069510265635750083533321030274383215).
Secret key calculation by Alice
  1: $S_K(x,y) = V_A \times P_B(x,y) =$ 123456789 * (1680504078792863419186290060061493676876263991152, 111209
     26205562069510265635750083533321030274383215) = (38027124171320004658630075130620602090153656563382,
     9358968692797266655333410538050707320151209943680).
Secret key calculation by Bob
  1: $S_K(x,y) = V_B \times P_A(x,y) =$ 987654321 * (41598408633041765117904814957892579387787288121723,
     30331027768661217591203589322093387887193929936089) = (38027124171320004658630075130620602090153656563382,
     9358968692797266655333410538050707320151209943680).

In this way, both parties get same secret key i.e., $S_K(x,y)$. In this illustration, 1% of the abscissa (i.e., x coordinate)
of $S_K$(x, y) is used in encoding and decoding of points in elliptic curve.
Encryption of plain OTP by Alice using public key of Bob
  1: Considers a plain OTP message as 32145688.
  2: Encodes the plain message into encoded message points in the elliptic curve using Koblitz algorithm as shown
     in Table 2 and in Figure 2.
  3: Encrypts the encoded message points into cipher message points as shown in Table 3 and in Figure 3, and sends
     the cipher message points to Bob.
Decryption by Bob using his own private key
  1: Decrypts cipher message points into encoded message points as shown as in Table 2 and in Figure 2.
  2: Decodes the encoded points into a plain message.
  3: Gets a plain message as 32145688.
```

Table 2: Encoded message points in the elliptic curve

SN	Pmsg(X)	Pmsg(Y)
1	1022	39724063396011179241991481543167237935714510129110
2	1001	3579332311729321835757266059425508498858988841211
3	981	45449421598805694936939313852002285339988567792348
4	1042	41905338400807894834998928213746564044963682008534
5	1064	30965928523274769546027471014828764333153039292943
6	1087	20058071397683232973900828570537209896964825497709
7	1122	4867272454768617028599194841722091173146360919109
8	1122	4867272454768617028599194841722091173146360919109

Table 3: Cipher points in the elliptic curve

SN	Cmsg(X)	Cmsg(Y)
1	30318105437745412012707811898660760983384011110715	11569707491906706117423094024694420747299985223720
2	52205651929554496519639339221161550504134904297868	40711808994104162900744001367694964600768099749387
3	40637385264950590178626612461851519009171269041008	21269888175723473719921751261815783080181718656214
4	48019670987601377650508574338611556007493705143248	46369536477813844035369879645549778427095937332635
5	32485373975889099014357103603609901409984388889309	50315819695451675351182042838026436049795166563072
6	35555618705331793276953267802821986756883330299327	36569338896318126138600676039130873843552205572619
7	14573041686907539131994344789859946931735552341691	48684012376533243830342291639743567258549511821547
8	14573041686907539131994344789859946931735552341691	48684012376533243830342291639743567258549511821547

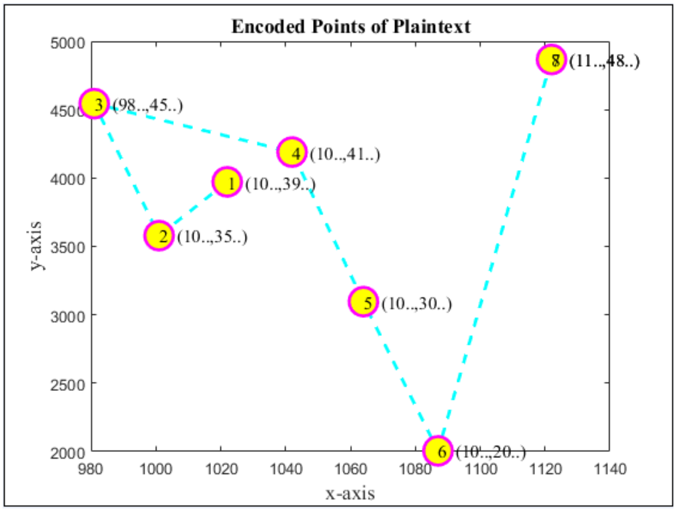


Figure 2: Encoded plain OTP before encryption

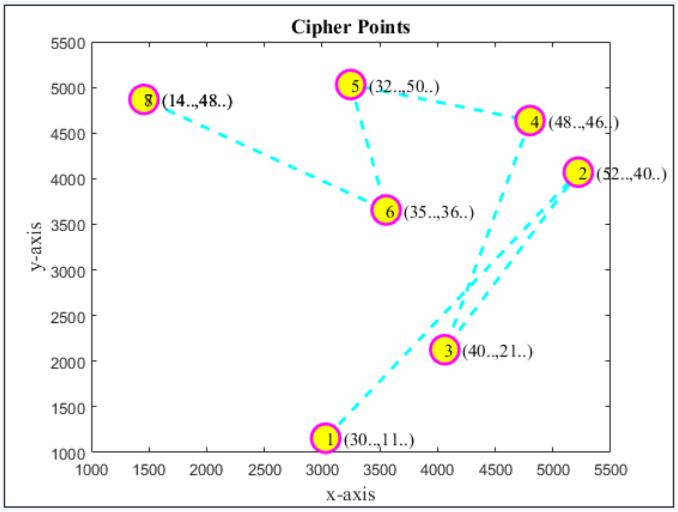


Figure 3: Cipher OTP points

5.2 Secrecy of Information

Case Study 1: Multi-authority Electronic Voting Scheme Based on Elliptic Curves by Porkodi *et al.* [35]. This paper proposed a security model for e-voting system, which works better with same parameters as used in DSA for building secured e-voting system. The paper also proposed that ECC needs considerably smaller parameters and provides the equivalent level of security as other asymmetric algorithms RSA and DSA which need much larger keys.

Case Study 2: Comparative Analysis of Public-Key Encryption Schemes by Alese *et al.* [1]. This research work focused on the comparative analysis of RSA encryption algorithm, ElGamal Elliptic Curve encryption algorithm, and Menezes-Vanstone elliptic curve encryption algorithm. These elliptic curve encryption schemes analog of ElGamal encryption scheme were implemented in Java, using the classes from the FlexiProvider library for RSA and ECC. Performance evaluation of the three algorithms based on the time lapse for their key generation, encryption, and decryption algorithms, and encrypted data size was carried out and compared. Their result confirmed that elliptic curve-based implementations are more superior to RSA-base implementations on all comparative parameters.

After comparing RSA and ECC ciphers, it has been proved that ECC involves much fewer overheads than RSA. ECC has many advantages due to its ability to provide the same level of security as RSA yet using shorter keys. However, its disadvantage which may even hide its attractiveness is its lack of maturity, as mathematicians, they believed that enough research has not yet been done in ECDLP.

Case Study 3: Nonce based ECC for Text and Image applications by Vigila *et al.* [48]. This paper implemented model based on ECC for text and image applications security. The paper suggested that ECC facilitates such as higher strength per bit leading to faster computation reduced power consumption and fewer storage requirements compared to RSA.

5.3 Authentication of Information

Case Study 1: Performance Comparison of Elliptic Curve and RSA Digital Signatures by Nicholas Jansma *et al.* [19]. This paper compared the performance characteristics of two public key cryptosystems (RSA and ECC) used in digital signatures to determine the applicability of each in modern technological devices and protocols that use such signatures.

Their findings suggest that for RSA key of size 1024-bit and greater, RSA key generation is significantly slower than ECC key generation. RSA is comparable to ECC for digital signature creation in terms of time and is faster than ECC for digital signature verification. Thus, for applications requiring message verification more often than the signature generation, RSA may be the better choice.

Case Study 2: A Secure and Efficient Remote User Authentication Scheme for Multi-server Environments Using ECC by Zhang, Junsong, *et al.* [49]. The paper presented that the requirements of operations are lesser in ECC-based than other related asymmetric-key schemes. That means that ECC requires less computational cost than other related public-key cryptosystems. The demonstration of the paper exhibits that proposed scheme can solve various types of security problems and is better suitable for memory-constrained devices.

Case Study 3: Chang *et al.* [8] proposed a strong RSA-based certificate-less signature scheme and claimed that their scheme is capable of resisting more intense malicious behavior.

Case Study 4: Sharma *et al.* [42] proposed an RSA-based efficient certificate-less signature scheme and proved that their scheme is safe under some well-studied assumptions. They also claimed that their scheme is suitable for WSN based on their implementation results on WSN.

Case Study 5: Deng *et al.* [11] proposed an identity-based proxy ring signature (IBPS) scheme using RSA without pairings, and used the random oracle model to prove the security of their scheme. They claimed that their scheme is more efficient than similar ones developed based on bilinear pairings.

Case Study 6: Singh *et al.* [43] experimentally evaluated the performance of digital signature signing and verification processes using RSA (Basic) and ECC. They claimed that RSA signature signing is slower than verification whereas ECC signature signing is generally faster than verification. This paper suggested that to use of ECC in place of RSA.

5.4 Implementing Hardware Security

Case Study 1: ECCs by Robshaw *et al.* [39]. In their paper, they provided a high-level comparison of RSA public-key cryptosystem and proposals for public-key cryptography based on elliptic curves.

There are however many issues to consider when making the choice between applications based on an elliptic curve cryptosystem and one based on RSA. In the paper, they have presented some of the issues (security, performance, standards and interoperability) that are perhaps most pertinent when making such a choice. The comparisons in this paper are made, however, under the premise that an elliptic curve cryptosystem over $GF(2^{160})$ offers the same security as 1024-bit RSA.

Case Study 2: Comparing ECC and RSA on 8-Bit CPUs by Gura *et al.* [14]. They proposed a new algorithm to reduce the number of memory accesses. Implementation and analysis led to three observations:

1) Public-key cryptography is viable on small devices without hardware acceleration. On an Atmel ATmega128 at 8 MHz, they measured 0.81s for 160-bit ECC point multiplication and 0.43s for a RSA-1024 operation with exponent $e = 2^{16} + 1$.
2) The relative performance advantage of ECC point multiplication over RSA modular exponentiation increases with the decrease in processor word size and the increase in key size.
3) Elliptic curves over fields using pseudo-Mersenne primes as standardized by NIST and SECG allow for high- performance implementations and show no performance disadvantage over optimal extension fields or prime fields selected specifically for a particular processor architecture.

They compared elliptic curve point multiplication over three SECG/NIST curves secp160r1, secp192r1, and secp224r1 with RSA-1024 and RSA-2048 on two 8-bit processor architectures. On both platforms, ECC-160 point multiplication outperforms RSA-1024 private-key operation by an order of magnitude and is a factor of 2 of RSA-1024 public-key operation. They presented a novel multiplication algorithm that significantly reduces the number of memory accesses. This algorithm led to a 25% performance increase for ECC point multiplication on the Atmel AVR platform. Their measurements and analysis led to fundamental observations: The relative performance of ECC over RSA increases as the word size of the processor decrease. This stems from the fact that the complexity of addition, subtraction and optimized reduction based on sparse pseudo-Mersenne primes grows linearly with the decrease of the word size whereas Montgomery reduction grows quadratically. As a result, ECC point multiplication on small devices becomes comparable in performance to RSA public-key operations and they expect it to be higher for large key sizes.

Case Study 3: Chatterjee *et al.* [9] focused on implementing an efficient architecture for scalar multiplication on binary Edwards curve in an analytical way and based on analytical and experimental results they claimed that their model helped in developing an architecture with improved efficiency in comparison to other similar models.

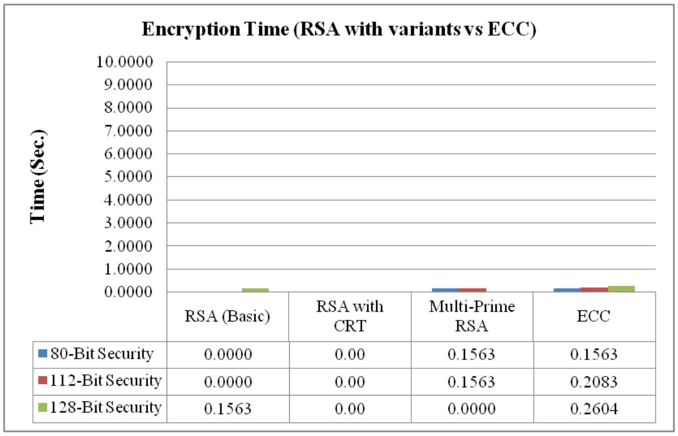
5.5 Wireless LAN Security

Case Study 1: Comparative Performance Analysis of Public-Key Cryptographic Operations in the WTLS Handshake Protocol by Rodriguez-Henriquez *et al.* [40]. They proposed a model for the protocol analysis considering the processing time of the cryptographic operations performed by the Client and the Server during the Negotiation protocol.

In their paper, an efficient realization of the WTLS (Wireless Transport Layer Security) handshake protocol was implemented on a realistic wireless scenario composed of a typical mobile device wirelessly connected to a workstation server. The data gathered in their experiments show that ECC consistently outperforms the traditional option represented by RSA in all the scenarios tested. Additionally, their analytical model predictions show a reasonable agreement with the obtained real data.

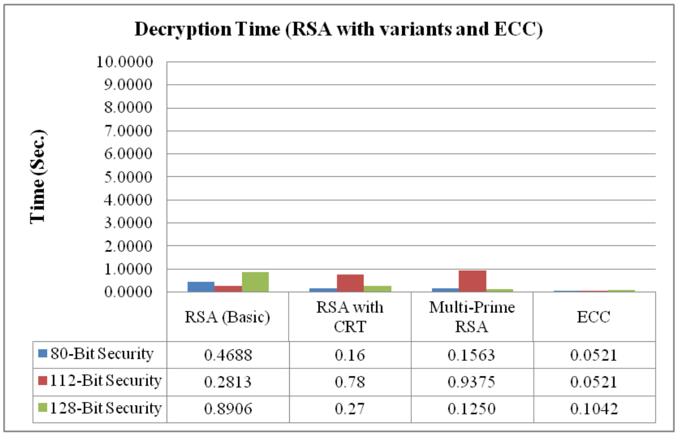
6 Performance Analysis of RSA and Its Two Variants with ECC

A performance analysis, based on encryption, decryption, and total time of RSA (Basic) with its two variants and ECC is mentioned here. The first variant of RSA is RSA with CRT and the second variant of RSA is the Multi-Prime RSA. For measuring time efficiency of these algorithms, the modulus used in experimentation are of 1024/2048/3072-bit for RSA and 160/224/256-bit for ECC, with two sample OTP message data of 27-bit (*i.e.*, "32145688") and 270-bit (*i.e.*, "OTP to transfer money to beneficiary A/C is "34741608". Do not share it with anyone"). Programs for these algorithms written and executed in C with GMP library, on Intel Pentium laptop with a dual-core processor (1.60 GHz, 533 MHz, 1 MB L2 cache), 2GB DDR2 RAM, under Ms-Windows platform. The performance analysis of RSA with its variants over ECC is shown in figures (Figure 4, 5, 6, 7, 8, 9. Upon experimentation, it is found that RSA (Basic) takes more time than its own two variations and ECC. ECC is better in terms of operational efficiency than RSA and its both variants as shown in Figure 6, and Figure 9.



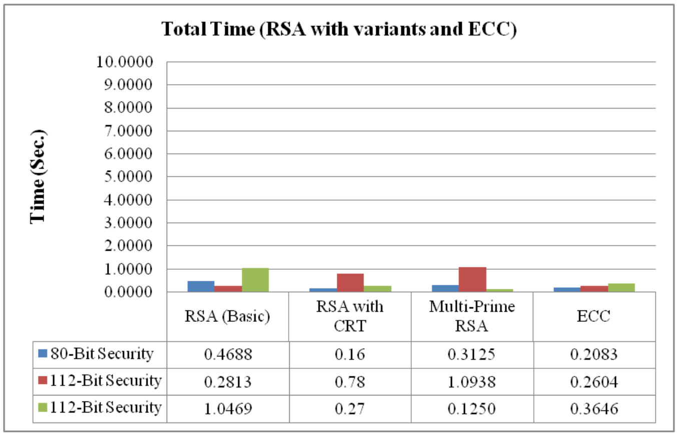
	RSA (Basic)	RSA with CRT	Multi-Prime RSA	ECC
80-Bit Security	0.0000	0.00	0.1563	0.1563
112-Bit Security	0.0000	0.00	0.1563	0.2083
128-Bit Security	0.1563	0.00	0.0000	0.2604

Figure 4: Encryption time (in seconds) of 27-bit data



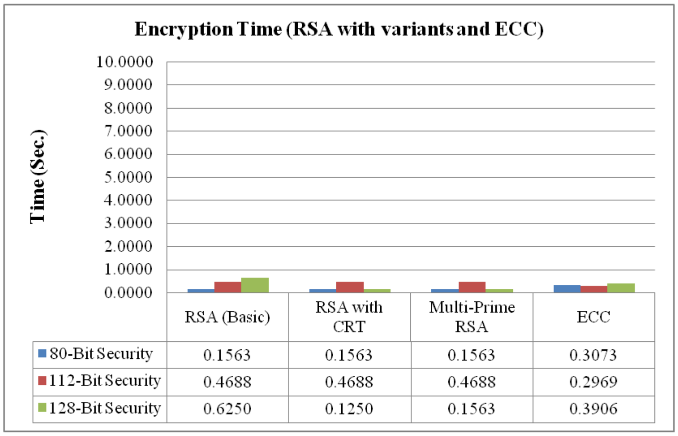
	RSA (Basic)	RSA with CRT	Multi-Prime RSA	ECC
80-Bit Security	0.4688	0.16	0.1563	0.0521
112-Bit Security	0.2813	0.78	0.9375	0.0521
128-Bit Security	0.8906	0.27	0.1250	0.1042

Figure 5: Decryption time (in seconds) of 27-bit data



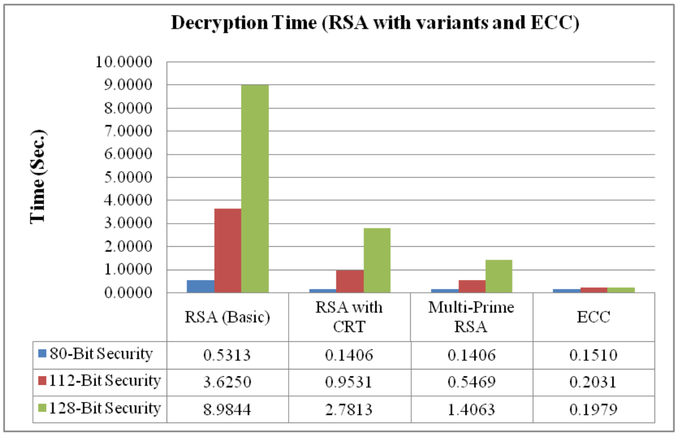
	RSA (Basic)	RSA with CRT	Multi-Prime RSA	ECC
80-Bit Security	0.4688	0.16	0.3125	0.2083
112-Bit Security	0.2813	0.78	1.0938	0.2604
112-Bit Security	1.0469	0.27	0.1250	0.3646

Figure 6: Total (Enc. and Dec.) time (in seconds) of 27-bit data



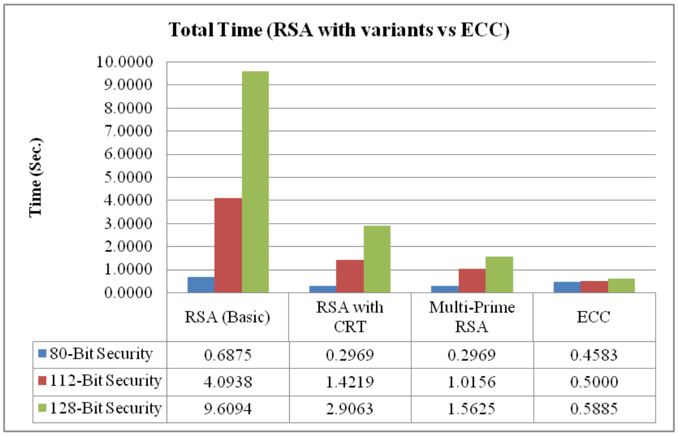
	RSA (Basic)	RSA with CRT	Multi-Prime RSA	ECC
🟦 80-Bit Security	0.1563	0.1563	0.1563	0.3073
🟥 112-Bit Security	0.4688	0.4688	0.4688	0.2969
🟩 128-Bit Security	0.6250	0.1250	0.1563	0.3906

Figure 7: Encryption time (in seconds) of 270-bit data



	RSA (Basic)	RSA with CRT	Multi-Prime RSA	ECC
🟦 80-Bit Security	0.5313	0.1406	0.1406	0.1510
🟥 112-Bit Security	3.6250	0.9531	0.5469	0.2031
🟩 128-Bit Security	8.9844	2.7813	1.4063	0.1979

Figure 8: Decryption time (in seconds) of 270-bit data



	RSA (Basic)	RSA with CRT	Multi-Prime RSA	ECC
🟦 80-Bit Security	0.6875	0.2969	0.2969	0.4583
🟥 112-Bit Security	4.0938	1.4219	1.0156	0.5000
🟩 128-Bit Security	9.6094	2.9063	1.5625	0.5885

Figure 9: Total (Enc. and Dec.) time (in seconds) of 270-bit data

7 Conclusion

Security of data communication is very important while data are being transmitted from one user to another user or system. Cryptography is one of the techniques to provide data communication security. This paper presented a performance study and an analysis of RSA (Basic) with its two popular variants and ECC. The experimental results for encryption, decryption and total time are taken by RSA with its variants and ECC are shown. It is concluded that ECC outperforms RSA and all the mentioned variants of RSA in terms of operational efficiency and security with lesser parameters. ECC with the Affine coordinate system is implemented here, the future research may implement the ECC with other than the Affine coordinate system to improve more efficiency of the ECC.

Acknowledgments

We would like to thank our colleagues, Head of Department of Computer Applications, Dean (R & C) and Director of our Institute for supporting directly or indirectly to this research work.

References

[1] B. K. Alese, E. D. Philemon and S. O. Falaki, "Comparative analysis of public-key encryption schemes," *International Journal of Engineering and Technology*, vol. 2, no. 9, pp. 1552–1568, 2012.

[2] E. Barker, W. Barker, W. Burr, W. Polk and M. Smid, "Recommendation for key management part 1: General (revision 3)," *NIST Special Publication*, vol. 800, no. 57, pp. 1–147, 2012.

[3] S. Basu, "A new parallel window-based implementation of the elliptic curve point multiplication in multi-core architectures," *Group*, vol. 14, pp. 101-108, 2012.

[4] K. Bhardwaj and S. Chaudhary, "Implementation of elliptic curve cryptography in c," *International Journal on Emerging Technologies*, vol. 3, no. 2, pp. 38–51, 2012.

[5] D. Boneh and H. Shacham, "Fast variants of RSA," *CryptoBytes*, vol. 5, no. 1, pp. 1–9, 2002.

[6] J. Bos, M. Kaihara, T. Kleinjung, A. K. Lenstra and P. L. Montgomery, *On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography*, Technical Report, 2009.

[7] C. C. Chang and M. S. Hwang, "Parallel computation of the generating keys for RSA cryptosystems," *Electronics Letters*, vol. 32, no. 15, pp. 1365–1366, 1996.

[8] C. C. Chang, C. Y. Sun and S. C. Chang, "A strong RSA-based and certificateless-based signature scheme," *International Journal Network Security*, vol. 18, no. 2, pp. 201–208, 2016.

[9] A. Chatterjee and I. Sengupta, “Performance modelling and acceleration of binary edwards curve processor on fpgas,” *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 80–93, 2015.

[10] T. Collins, D. Hopkins, S. Langford and M. Sabin, *Public Key Cryptographic Apparatus and Method*, Dec. 8, 1998. US Patent 5,848,159.

[11] L. Deng, H. Huang and Y. Qu, “Identity based proxy signature from RSA without pairings,” *International Journal Network Security*, vol. 19, no. 2, pp. 229–235, 2017.

[12] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[13] X. D. Dong, “A multi-secret sharing scheme based on the crt and RSA,” *International Journal of Electronics and Information Engineering*, vol. 2, no. 1, pp. 47–51, 2015.

[14] N. Gura, A. Patel, A. Wander, H. Eberle and S. C. Shantz, “Comparing elliptic curve cryptography and RSA on 8-bit cpus,” in *CHES*, vol. 4, pp. 119–132, 2004.

[15] L. Han, Q. Xie and W. Liu, “An improved biometric based authentication scheme with user anonymity using elliptic curve cryptosystem,” *International Journal Network Security*, vol. 19, no. 3, pp. 469–478, 2017.

[16] D. Hankerson, A. J. Menezes and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer Science & Business Media, 2006.

[17] G. Hou and Z. Wang, “A robust and efficient remote authentication scheme from elliptic curve cryptosystem,” *International Journal Network Security*, vol. 19, no. 6, pp. 904–911, 2017.

[18] M. S. Hwang, K. F. Hwang, I. C. Lin, “Cryptanalysis of the batch verifying multiple RSA digital signatures”, *Informatica*, vol. 11, no. 1, pp. 15-19, Jan. 2000.

[19] N. Jansma and B. Arrendondo, “Performance comparison of elliptic curve and RSA digital signatures,” *nicj. net/files*, 2004.

[20] N. Koblitz, “Elliptic curve cryptosystems,” *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[21] V. B. Kute, P. R. Paradhi and G. R. Bamnote, “A software comparison of RSA and ECC,” *International Journal Computer Science Applications*, vol. 2, no. 1, pp. 43–59, 2009.

[22] L. Liu, Z. Cao, W. Kong and J. Wang, “On bilinear groups of a large composite order,” *International Journal of Electronics and Information Engineering*, vol. 7, no. 1, pp. 1–9, 2017.

[23] E. J. L. Lu, M. S. Hwang, and C. J. Huang, “A new proxy signature scheme with revocation”, *Applied Mathematics and Computation*, vol. 161, no. 3, PP. 799-806, Feb. 2005.

[24] D. Mahto, D. A. Khan and D. K. Yadav, “Security analysis of elliptic curve cryptography and RSA,” in *Proceedings of the World Congress on Engineering*, vol. 1, 2016.

[25] D. Mahto and D. K. Yadav, “Network security using ecc with biometric,” in *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pp. 842–853, 2013.

[26] D. Mahto and D. K. Yadav, “Enhancing security of one-time password using elliptic curve cryptography with biometrics for e-commerce applications,” in *Third International Conference on Computer, Communication, Control and Information Technology (C3IT'15)*, pp. 1–6, 2015.

[27] D. Mahto and D. K. Yadav, “Enhancing security of one-time password using elliptic curve cryptography with finger-print biometric,” in *2nd International Conference on Computing for Sustainable Global Development (INDIACom'15)*, pp. 1737–1742, 2015.

[28] D. Mahto and D. K. Yadav, “Security improvement of one-time password using crypto-biometric model,” in *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*, pp. 347–353, 2016.

[29] D. Mahto and D. K. Yadav, “One-time password communication security improvement using elliptic curve cryptography with iris biometric,” *International Journal of Applied Engineering Research*, vol. 12, no. 18, pp. 7105–7114, 2017.

[30] D. Mahto and D. K. Yadav, “Rsa and ECC: A comparative analysis,” *International Journal of Applied Engineering Research*, vol. 12, no. 19, pp. 9053–9061, 2017.

[31] D. Mahto and D. K. Yadav, “Secure online medical consultations using elliptic curve cryptography with iris biometric,” *International Journal of Control Theory and Applications*, vol. 10, no. 13, pp. 169–179, 2017.

[32] A. Mansour, A. Davis, M. Wagner, R. Bassous, H. Fu and Y. Zhu, “Multi-asymmetric cryptographic RSA scheme,” in *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, p. 9, 2017.

[33] V. S. Miller, “Use of elliptic curves in cryptography,” in *Conference on the Theory and Application of Cryptographic Techniques*, pp. 417–426, 1985.

[34] V. S. Naresh and N. V. Murthy, “Elliptic curve based dynamic contributory group key agreement protocol for secure group communication over ad-hoc networks,” *International Journal Network Security*, vol. 17, no. 5, pp. 588–596, 2015.

[35] C. Porkodi, R. Arumuganathan and K. Vidya, “Multi-authority electronic voting scheme based on elliptic curves.,” *International Journal Network Security*, vol. 12, no. 2, pp. 84–91, 2011.

[36] Q. Qian, Y-L Jia and R. Zhang, “A lightweight rfid security protocol based on elliptic curve crytography,” *International Journal Network Security*, vol. 18, no. 2, pp. 354–361, 2016.

[37] J. J. Quisquater and C. Couvreur, "Fast decipherment algorithm for RSA public-key cryptosystem," *Electronics letters*, vol. 18, no. 21, pp. 905–907, 1982.

[38] R. L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[39] M. J. B. Robshaw and Y. L. Yin, "Elliptic curve cryptosystems," *An RSA Laboratories Technical Note*, vol. 1, p. 997, 1997.

[40] F. Rodrguez-Henrquez, C. E. Lpez-Peza, M. A. Len-Chvez and P. Puebla, "Comparative performance analysis of public-key cryptographic operations in the wtls handshake protocol," in *Proceedings of the 1st International Conference on Electrical and Electronics Engineering*, pp. 24–27, 2004.

[41] K. R. Santosh, C. Narasimham and P. Shetty, "Cryptanalysis of multi-prime RSA with two decryption exponents," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 40–44, 2016.

[42] G. Sharma, S. Bala and A. K. Verma, "An improved RSA-based certificateless signature scheme for wireless sensor networks," *International Journal Network Security*, vol. 18, no. 1, pp. 82–89, 2016.

[43] S. R. Singh, A. K. Khan and S. R. Singh, "Performance evaluation of RSA and elliptic curve cryptography," in *2nd International Conference on Contemporary Computing and Informatics (IC3I'16)*, pp. 302–306, 2016.

[44] M. S. Srinath and V. Chandrasekaran, "Isogeny-based quantum-resistant undeniable blind signature scheme," *International Journal of Network Security*, vol. 20, no. 1, pp. 8–17, 2018.

[45] A. Takayasu and N. Kunihiro, "General bounds for small inverse problems and its applications to multi-prime RSA," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 100, no. 1, pp. 50–61, 2017.

[46] S. F. Tzeng, M. S. Hwang, "Digital signature with message recovery and its variants based on elliptic curve discrete logarithm problem", *Computer Standards & Interfaces*, vol. 26, no. 2, pp. 61–71, Mar. 2004.

[47] P. Verma, D. Mahto, S. K. Jha and D. K. Yadav, "Efficient RSA cryptosystem with key generation using matrix," *International Journal of Control Theory and Applications*, vol. 10, no. 13, pp. 221–228, 2017.

[48] S. M. C. Vigila and K. Muneeswaran, "Nonce based elliptic curve cryptosystem for text and image applications.," *International Journal Network Security*, vol. 14, no. 4, pp. 236–242, 2012.

[49] J. Zhang, J. Ma, X. Li and W. Wang, "A secure and efficient remote user authentication scheme for multi-server environments using ECC," *TIIS*, vol. 8, no. 8, pp. 2930–2947, 2014.

Biography

Dindayal Mahto is a Faculty Member cum Ph. D. Research Scholar in the Department of Computer Applications at National Institute of Technology Jamshedpur, India. He received his M. Tech. degree in Information Security from Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India in 2012. His research interests include cryptography, information/biometric security.

Dilip Kumar Yadav is an Associate Professor in the Department of Computer Applications at National Institute of Technology, Jamshedpur, India. He received his Ph. D. degree in software reliability engineering from Indian Institute of Technology, Kharagpur (India) in 2012. He received the B. Tech. (ME) and M. Tech. (CIDM) degrees from National Institute of Technology, Jamshedpur, India, in 1991 and 1994 respectively. His research interests include software reliability and quality modeling, software security, soft computing and system optimization.