ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
**Indian Journal of Science and Technology,** *Vol 11(17), DOI: 10.17485/ijst/2018/v11i17/122766, May 2018*

# Multi Key Generation Scheme for Cloud and IoT Devices

**A. Nithya, S. Dhivya and T. Abirami**

Department of Electronics and Communication Engineering, M. Kumarasamy College of Engineering, Karur – 639113, Tamil Nadu, India; nithyaa.ece@mkce.ac.in, dhivyas.ece@mkce.ac.in, abiramit.ece@mkce.ac.in

## Abstract

**Objective**: A brand new version of RSA, known as Multi Key Generation Scheme generation scheme, has been proposed. For sensitive information, our schemes will be a useful resource in replacing the records among cloud to IoT and IoT devices. Whilst cryptography belongs to the asymmetric type, then it has public and personal keys. **Methods/Statistical Analysis:** For reminiscence performance, our scheme reuses the RSA scheme with a Diophantine shape of the nonlinear equation. Furthermore, our scheme overall performances well and this is especially because of the use of RSA public key. **Findings**: MKGS does not require multiplicative inverse function or prolonged Euclid's algorithm. Finally, we've made an experimental result on diverse stages of MKGS P.C such as key generation, encryption, and decryption via varying the N-bit modulo bits from 1K to 10K. **Application/Improvements:** We can develop new features rather than using Modulation bits.

**Keywords:** Cloud computing, Cryptography, Multi Key, Public Key Cryptosystem, IoT, RSA

## 1. Introduction

A creation Cryptography is widely labeled as Symmetric, Asymmetric and Hybrid[1]. Whilst cryptography belongs to the uneven type, then it has public and private keys. Presently, Public Key Cryptography (%)[2–6] plays a crucial position in numerous areas which includes Banking, on-line buying and e-mail. Due to this, there may be the excessive risk of having attack[7] via by guessing the well-known RSA mystery keys from the general public exponent. some of the recent versions of RSA with respect to their overall performance evaluation[8–14] and memory constraints of key. Some of the P.C are ideal for multi-key generation scheme for green sharing of facts among the entities like IoT and Cloud computing. Right here we've got analyzed the multi-key based totally cryptosystems with reuse of keys are as follows: ESRKG (improved and Secured RSA primarily based Key era)[15], dual RSA[16], Trivial RSA[17], and N-high RSA[18]. In those variations, the power trivially depends at the N-bit moduli and due to this the time reminiscence tradeoff additionally receives extended. But the IoT- based totally device[19] has the minimal hardware constraint inclusive of low strength and low computation of round 2K bits. For achieving excessive-security strength with low moduli bits, we endorse here MEMK scheme through RSA public key20 thing, NE and Diophantine co-ordinates. The same old Diophantine to comprehend our proposed MEMK scheme, we present the primary workflow as follows: to begin with, the cloud generates the multiple keys to its IoT gadgets via selecting the correct case. The general public key is shared by using cloud with the IoT sensor or data transmitter, with the

*Author for correspondence

intention to transfer the sensitive sensor facts by means of shape of equation additionally called as Pell's that's inside the shape of 22 1 ii X RY −=and by way of fixing this we able to get the co-ordinates, ii XY. In our scheme, we have 3 instances of producing the multi-keys as follows: Case: 1 by way of preserving the RSA parameters ( ), NE φ as steady after which various the Diophantine co-ordinates, ii XY Case: Case: 2 by means of retaining the Diophantine coordinates, ii XY constant after which varying RSA parameters ( ), NE φ; Case: 3 by way of various both Diophantine coordinates, ii XY and RSA parameters ( ), NE φ it's miles found that one may obtain *nm number of IoT keys for secure transportation; here 'n' is the variety of RSA public pairs and 'm' is the range of Diophantine co-ordinates.

Changing from readable shape to cipher shape referred to as encryption to the opposite end known as cloud or to IoT receiver. After receiving this cipher from either the cloud or the IoT receiver wishes the non-public key to convert from cipher to readable form and this method is called decryption. The corresponding private key is likewise shared by using the cloud to the corresponding IoT tool. The fundamental workflow of our proposed MEMK version is proven in Figure 1. The relaxation of this work is prepared as: First, we describe the P.C advent in phase 1. Next, in phase 2, we make a related scheme such Std. RSA, ESRKGS, twin RSA, Trivial RSA and N-prime RSA. In segment 3, we present our proposed scheme MEMK scheme. Section four describes the numerical effects of our proposed scheme in conjunction with related %'s. section 5 suggests the comparative consequences of overall performance on many stages. Ultimately, we can give a conclusion in section 6. II. Related WORKS in this phase, the RSA variants of uneven cryptosystems are going to be reviewed as it is. Some of the current versions of RSA-P.C's are Std. RSA, ESRKGS, twin RSA, and Trivial RSA.

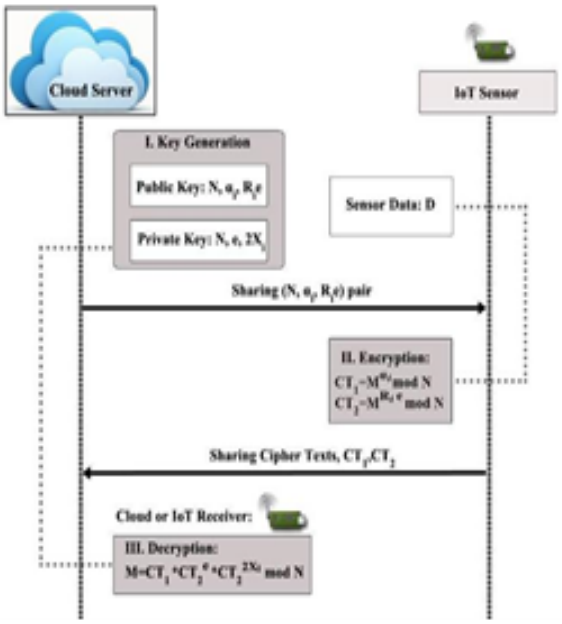


**Figure 1.** MEMK model processing diagram.

## 1.1 Standard RSA

This well-known globally relevant technique is first proposed by way of the eminent researcher's named[10] for to have each cozy conversation and acquiring virtual signatures on E-commerce primarily based applications. This scheme carries three most important phases which include key era; encryption and decryption are as follows:

1. Choose any two distinct prime numbers p and q. The integer's p and q values would be chosen at random.

2. Compute n=p*q; 'n' act as the modulus value of both public and private keys.

3. Compute Euler's Totient Function, Φ (n) = (p - 1)*(q - 1)

4. Choose an integer **'e'** such that 1<e< and GCD (e,) =1, i.e., e and Φ (n) are relatively prime. Hence 'e', is released as public key exponent.

5. Determine d = $e^{-1}$ (mod Φ (n)), which implies (d * e) = 1(mod Φ(n)) here d lies in (0≤d≤n)

6. Encryption: Obtain the Cipher Text from the message, C (M) = $M^{e}$ mod $n^{e}$

7. Decryption: Obtain the message from Encrypted Text, M (C) = $C^{d}$ mod $nc^{e}$

## 1.2 ESRKGS

More suitable and Secured RSA KG Scheme) this approach extends the RSA percent by using taking 4 primes with a couple of Euler totient function. The last goal of this machine is putting off brief public exponent assaults like Wiener or Coppersmith assaults. Even then the safety of ESRKGS is going with Std. RSA, because it uses small private'd' for decryption. Eventually, the complexity of ESRKGS personal key guessing assault is equivalent to Std. RSA. Some of the famous personal key attacks may be extracted via a factorization method like Fermat or Shor's algorithm *Take four large prime numbers p, q, r, s.*

1. Compute n=p*q and m=r*s
2. Compute, N=m*n.
3. Compute Euler phi value of n and m, ϕ(n)=(p-1)*(q-1), ϕ(m)=(r-1)*(s-1)
4. Compute $\phi$(N)=ϕ(n)*ϕ(m).
5. Find a random number e1, satisfying 1< e1 <(n) and gcd(e1, ϕ(n) )=1.
6. Find a random number e2, satisfying 1< e2 <(n) and gcd(e2, ϕ(n) )=1.
7. Compute E1=e1 mod N.
8. Find Random number E, satisfying 1<E<ϕ(N) *E1 and gcd (E, ϕ(N)* E1)=1
9. Compute a random number D, such that, D=$E^{-1}$mod (ϕ (N)*E1)
10. Encryption: $C_e$ (M) = $M^{E}$ mod n
11. Decryption: $M_d$ ($C_e$) = $C^{D}$ mod n

## 1.3 Dual RSA

Here the 2 communication parties can have the distinct non-public keys and moduli by sharing the same public exponent.

1.There exists two tremendous integer k1 and k2 such ed = 1 + k1 and ed = 1 + k2 Ø(N2).

2.Select any four wonderful top numbers (p1, q1) and (p2, q2)

3.Compute the moduli N1=p1q1 and N2=p2q2

4.Now compute the Euler characteristic, Ø(N1) = ( p1 - 1)*( q1 - 1) & Ø(N2). = ( p2 - 1)*( q2 - 1)

5.Pick out an integer 'e' such that Ø(N1) <e< Ø(N2). And GCD (e, Ø(N1), Ø(N2).) =1 is enormously prime. Wherein, e, N1, N2, is released as a public key exponent.

6.Now determine 'd1 and d2'

Worldwide convention on improvements in electricity and advanced Computing technologies [i-PACT2017] three where, ed1≡1 (mod Φ (N1)) & also ed2≡1 (mod Φ (N2)), here d1 lies in (zero ≤d1 ≤N1) & d2 lies in zero≤ d2 ≤N2). Here the personal key exponents are as follows: (d1, d2, p1, q1, p2, q2)

## 1.4 Trivial RSA

Here the 3 communication events can have the distinct personal keys and moduli by way of sharing the equal public exponent.

1.Randomly select $a_1$, $a_2$ such that $r_1$= $a_1$ $a_{2+}$ 1 where $r_1$ is prime.

2.Randomly select $b_2$ such that $r_2$ = $a_1$ $b_2$ + 1 where $r_2$ is prime.

3.Randomly select $b_1$ such that $r_3$= a $_2$ $b_1$ + 1 where $r_3$ is prime.

4.Now compute the following primes $s_1$ , $s_2$ , $s_3$ ,
$s_1 = b_1 b_2 + 1$, $s_2 = z1\ a_2 + 1$, $s_3 = z1\ a_1 + 1$, $z_2 = b_1$,
$z_2 = b_1$,
$z_3 = b_2$, $ED = 1 + z_1 \varphi N_1 = 1 + z_2 \varphi N_2 = 1 + z_3 \varphi N_3$,
where, $N_i = a_i\ b_i$

5. Compute D by using above step 3.
Output: Public key E, $N_1$, $N_2$ , $N_3$ and Private Key D, $r_1$, $r_2$ , $r_3$, $s_1$ , $s_2$ , $s_3$

6. Encryption:
Compute, $C_i = M_i^E \bmod N_i$ 7. Decryption:
Compute, $P_i = C_i^D \bmod N_i$

### 1.5 N-prime RSA

1. Here the N-bit modulus is generated by deciding on greater than primes. *Choose two or more distinct prime numbers p, q, r and so on values would be chosen at random.*

2. Compute N=p*q*r; 'N' act as the modulus value of both public and private keys.

3. Compute Euler's Totient Function,
Φ (N) = (p - 1)*(q - 1) * (r - 1) and so on.

4. Compute Euler's Totient Function, Φ (n) = (p - 1)*(q - 1)

5. Select a public key component 'e' such that 1<e< and GCD (e, ) =1. Hence 'e', is released as public key exponent.

6. for i=1 to N do

7. $\alpha i = [yi + \Phi(n)]^2 - P[xi + e]^2$

8. end for

9. Encryption:
Obtain the Cipher Text from message M,
***For IoT1:*** $M0^{\alpha 1} \bmod N = CTi1$
$M0^{pe} \bmod N = CTi2$
***For IoTn:*** $Mn^{\alpha 1} \bmod N = CTn1$
$Mn^{pe} \bmod N = CTn2$

10. Decryption: Obtain the message from Encrypted Text CTi, $(CT1 * CT2^e * CT2^{2Xi}) \bmod N = M$

## 2. Proposed Multikey

Technology version *Cloud Key Generation initializes all the key variables.*

1. Choose any two distinct random prime p and q.

2. Compute N=p*q; Here 'N' act as the modulus value for both public and private key.

3. Compute Euler's Totient Function, Φ (n) = (p - 1)*(q - 1)

4. Select a public key component 'e' such that 1<e< and GCD (e, ) =1. Hence 'e', is released as public key exponent.

5. For i=1 to N do

6. $7.\alpha i = [yi + \Phi(n)]^2 - P[xi + e]^2$

7. End for

8. Encryption: Obtain the Cipher Text from message M,

***9. For IoT1:*** $M0^{\alpha 1} \bmod N = CTi1$
$M0^{pe} \bmod N = CTi2$
***For IoTn:*** $Mn^{\alpha 1} \bmod N = CTn1$
$Mn^{pe} \bmod N = CTn2$

10. Decryption: Obtain the message from Encrypted Text CTi, $(CT1 * CT2^e * CT2^{2Xi}) \bmod N = M$

## 3. Overall Performance

Evaluation of the numerous phases of %'s includes key generation, encryption, and decryption time using i5

**Table 1.** Key generation time

| N-bit Moduli | RSA | KGS Time (ms) | | MEMK |
|---|---|---|---|---|
| | | ESR | N Prime | |
| 1024 | 200 | 404 | 459 | 117 |
| 2048 | 693 | 2744 | 1643 | 610 |
| 3072 | 2215 | 9172 | 2989 | 1324 |
| 4096 | 8194 | 35987 | 10065 | 3265 |
| 5120 | 14948 | 74753 | 18018 | 6348 |
| 6144 | 44204 | 194842 | 57181 | 1359 |
| 7168 | 73783 | 222204 | 86880 | 2367 |
| 8192 | 105532 | 337757 | 149887 | 47843 |
| 9216 | 129518 | 515191 | 165973 | 84321 |
| 10240 | 338383 | 619289 | 392465 | 169029 |

**Table 2.** Encryption time

| N-bit Moduli | RSA | KGS Time (ms) | | MEMK |
|---|---|---|---|---|
| | | ESR | N Prime | |
| 1024 | 30 | 42 | 37 | 39 |
| 2048 | 43 | 33 | 63 | 92 |
| 3072 | 63 | 98 | 128 | 347 |
| 4096 | 114 | 64 | 274 | 773 |
| 5120 | 188 | 118 | 428 | 1274 |
| 6144 | 387 | 174 | 691 | 2437 |
| 7168 | 485 | 266 | 972 | 3491 |
| 8192 | 749 | 385 | 1401 | 5742 |
| 9216 | 1055 | 45792 | 2726 | 2138 |
| 10240 | 1489 | 86767 | 5691 | 45331 |

**Table 3.** Decryption Time

| N-bit Moduli | RSA | KGS Time (ms) | | MEMK |
|---|---|---|---|---|
| | | ESR | N Prime | |
| 1024 | 8 | 6 | 19 | 12 |
| 2048 | 27 | 16 | 113 | 14 |
| 3072 | 80 | 120 | 336 | 15 |
| 4096 | 184 | 98 | 770 | 16 |
| 5120 | 353 | 202 | 1885 | 17 |
| 6144 | 763 | 315 | 2324 | 16 |
| 7168 | 938 | 500 | 3947 | 15 |
| 8192 | 1470 | 713 | 5911 | 16 |
| 9216 | 2094 | 739 | 9990 | 19 |
| 10240 | 2897 | 2719 | 19767 | 21 |

Intel core processor with 8GB RAM on windows 8 64-bit platform. By way of various the bit-lengths of N-bit moduli from 1K to 10K the outcomes are proven within the following desk I to III. Here the N-bit moduli trivially rely upon the high choice bit duration, which randomly selects using the Java Big Integer possibly top function and for timing dimension gadget. Current Time Millis feature is used. For all the RSA variations the usual public issue is constantly taken as 216+1. The % levels of these variants are cautiously study with the aid of generating hundred random samples. In Table 1, it is very clear that our Cloud multi-Key generation system has performed well when compared to its breeds.

Whereas in Table 2, it has been noticed that our proposed algorithm takes more time tradeoff, mainly due to double encryption of IoT sensor sensitive data. For lightweight cryptography, this can also be improved by choosing the *short* RSA public exponent 'e' for the generation of MEMK multi keys. Since our MEMK scheme takes the public exponent as i and *P .e* for cipher text's *CT1* and *CT2* production. Table 3 shows the decryption time of various schemes. In Table 3, it has been noticed that our proposed scheme MEMK outperformed well by taking only *12ms* to *21ms* for the input moduli *N* from *1K* to *10K*. From the numerical result, it has been very clear that our proposed scheme is well suited to real-time applications of IoT devices.

## 4. Comparative Analysis

Here, the overall time of our proposed MEMK scheme and its variants are presented in the above Figure 2. It is very clear that our proposed scheme leads an efficient time trade-off when compared to its variants. Hence our scheme is well suited to it or light weight based devices.

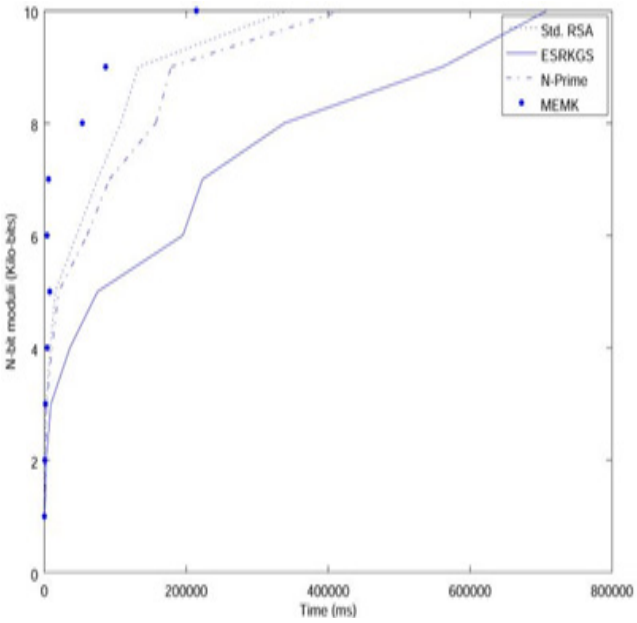


**Figure 2.** Comparative analysis.

## 5. Conclusion

We have given our MEMK PKC with its execution measure of RSA, *N*-prime, and ESR in section IV. From the KGS results at Table 1, our MEMK takes just 0.50% of RSA,0.43% of N-prime, and 0.27% of ESR. From the Encryption results at Table 2, our MEMK takes only 0.52% of ESR. From the Encryption results at Table 3, our MEMK takes simply 0.007% of RSA, 0.001% of *N*-Prime, and 0.008% of RSA.

Hence, is very clear that our MEMK scheme is well suited to real-time applications of IoT or light weight devices. Moreover, our scheme generated multiple keys with effective time-memory trade-off. In future, the cryptography keys will be highly reused to optimize the memory space of the cloud. Our MEMK algorithm also supports dynamic change of keys under various constraints of cluster sharing domains.

## 6. References

1. Chang CC, Hwang MS. Parallel computation of the generating keys for RSA cryptosystems. IET Journals & Magazines, Electronics Letters. 1996; 32(15):1365–6. Crossref.
2. Thirumalai CS, Senthilkumar M, Silambarasan R, Westphall CB. Analyzing the strength of Pell's RSA. International journal of Pharmacy and Technology. 2016; 8(4):21869–74.
3. Thangavel M, Varalakshmi P, Murrali M, Nithya K. An Enhanced and Secured RSA Key Generation Scheme (ESRKGS). Journal of information Security and application. 2015; 20(2):3–10.
4. Thirumalai CS, Senthilkumar M, Vaishnavi B. Physicians Medicament using Linear Public Key Crypto System. International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT); 2016. p. 1936–9. Crossref.
5. Emanuele B, Murru N. An efficient and secure RSA--like cryptosystem exploiting R'edei rational functions over Conics; 2015. p. 1–18.
6. Thirumalai CS. Review on the memory efficient RSA variants. International Journal of Pharmacy and Technology. 2016; 8(4):4907–16.
7. Hung-min S, Mu-en W, Wei-chi T, Jason HM. Dual RSA and its security analysis, IEEE transactions on information theory. 2007, 53 (8):2922-2933. Crossref.
8. Viswanathan P. Fusion of cryptographic watermarking medical image system with reversible property. Computer Networks and Intelligent Computing; 2011. p. 533–40.
9. Segar TC, Vijayaragavan R. Pell's RSA key generation and its security analysis. Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT); 2013. p. 1–5. Crossref.
10. Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM. 1978; 212):120–6. Crossref.
11. Thirumalai CS. Physicians Drug encoding system using an Efficient and Secured Linear Public Key Cryptosystem (ESLPKC). International Journal of Pharmacy and Technology. 2016; 8(3):16296–303.
12. Jhalani M, Singh P, Shrivastava G. Enhancement over the variant of public key cryptography algorithm. In International Journal of emerging Technology and Advanced Engineering. 2012; 2(12).
13. Chandramowliswaran NS, Srinivasan S. Muralikrishna P. Authenticated key distribution using given set of primes for secret sharing. Systems Science & Control Engineering. 2015; 3(1):106–12. Crossref.
14. Chhabra A, Mathur S., 2011, Modified RSA algorithm: A secure approach. International Conference on Computational Intelligence and Communication Networks; 2011. p. 545–8. Crossref.

15. Chandramowliswaran N, Srinivasan S, Chandra ST. A note on linear based set associative cache address system. International Journal on Computer Science and Engineering. 2012; 4(08):1383–6.
16. Forouzan BA. Cryptography and network security. Special Indian Edition. Tata McGraw-Hill; 2007. PMid:17450327
17. Thirumalai CS, Senthilkumar M. Secured E-mail system using base 128 encoding scheme. International Journal of Pharmacy and Technology. 2016; 8(4):21797–806.
18. Dhakar RS, Gupta AK, Sharma P. Modified RSA Encryption Algorithm (MREA). Second International Conference on Advanced Computing & Communication Technologies; 2012. p. 426–9.
19. Chandramowliswaran N, Srinivasan S, Segar TC. A novel scheme for secured associative mapping. The International Journal of Computer Science and Applications. 2012; 1(5):1–7
20. Debiao H. Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. IEEE Transactions on Information Forensics and Security. 2016; 11(9):2052–64. Crossref.