ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

An Implementation of Securing the Sensitive Data at Application Level using P^MQ^N-RSA in Big Data

Naveen Kumar R[#1], Prof Padmavathamma Mokkala[*2]

Research Scholar, Department of Computer Science, S.V.University, Tirupathi, India[#1]

BOS, Department of Computer Science, S.V.University, Tirupathi, India[*2]

ABSTRACT: An era has come where Organizations and individuals are more connected to digitally than ever before. As a significance, data is accessible to more people than ever before. While digitization accelerates information sharing of sensitive, it exacerbates the threat of sensitive data falling into the un-authorized / wrong hands. To combat this sensitive data threat, enterprises turn to cryptosystem. In the cryptosystem encryption is the process of encoding sensitive data so that only authorized or privileged parties can decrypt and read the sensitive data applying this methodology in application level we provide complete security on the sensitive data.

KEYWORDS: Big Data; Sensitive Data; Application Level; RSA; P^MQ^N-RSA; cryptosystem; authorization; Privilege users

I. INTRODUCTION

In the modern distributed era has come where organizations and individuals are more connected to digitally than ever before. In the Digital world the government\Companies collecting the massive data of their resource\people. For marketing and research, many of the businesses uses this big data, but may not have the fundamental assets particularly from a security perspective. If a security breach occurs to big data, it would result in even more serious legal repercussions and reputational damage than at present.

In this new era, many companies are using the technology to store and analyze petabytes of data about their data, business of their customers\people. As a result, information classification becomes even more critical. In Government\organizations classification of sensitive data and encrypting the sensitive data is very essential. Not only security but also data privacy challenges existing industries and federal organizations. With the increase in the use of big data in business, many companies are wrestling with privacy issues on the sensitive data. Data privacy [1][2] is a liability, thus Government\companies must be on privacy defensive on sensitive data. But unlike security, privacy on sensitive data should be considered as an asset. There should be a balance between data privacy and national security on sensitive data.

II. RELATED WORK

Data sources for information fed into a Big Data implementation inevitably contain either sensitive, protected information or key intellectual property. This information is distributed throughout the Big Data implementation. That entire sensitive data should be protected.

Today’s big data environments often include both sensitive and no sensitive data (including anonymous data). Hackers can correlate de-anonymized data sets to identify people and their preferences. Generally speaking, outsiders are prevented from accessing big data environments by traditional perimeter security at the boundaries of a private network. However, with today’s sophisticated break-in strategies, perimeter security is no longer adequate. Criminals

DOI: 10.15680/IJIRCCE.2017. 0503115

often try to lift health information, credit card numbers, and other vital information in order to sell it on the black market. No company wants its data to be compromised or its systems to be breached. However, most traditional IT security practices aren't strong enough to resist the new types of malware, phishing schemes, netbots, and SQL injection attacks unleashed by cybercriminal organizations for sensitive data.

Security[7][8][9] Issues with Hadoop Many of today's big data projects incorporate Apache Hadoop, an open-source framework for storing and processing big data in a distributed fashion. Business analysts load data into Hadoop to detect patterns and extract insights from structured, semi-structured, and unstructured data. Unfortunately, not all organizations have strong data security in place for these activities. There may be personally identifiable information and intellectual property loaded into these data sets. Initially developed as a way to distribute big data processing jobs among many clustered servers, the Hadoop architecture wasn't built with security in mind. Namely, it lacks access controls on the data, including password controls, file and database authorization, and auditing. As such, it doesn't comply with important industry standards such as the Health Insurance Portability and Accountability Act (HIPAA) and the Payment Card Industry Data Security Standard (PCI DSS). Sometimes supplementary group of users can access sensitive data. So we need to provide the privileges user can access sensitive data.

For many organizations big data has evolved into an enterprise data platform. That poses new security challenges as data that was once siloed is brought together in a vast data lake and made accessible to a variety of users across the organization. Among these challenges are:

- Ensuring that authorized users can only access the sensitive data that they are entitled to access.
- Ensuring the protection of data—both at rest and in transit—through enterprise-grade encryption.

III. PROPOSED ALGORITHM

In our approach our secure model will provide government\organizations can restrict the sensitive data access and data theft which leads potential threat of the government\organization. To overcome this issue we are proposing the privilege access control on sensitive data of the user's in application level. Sensitive data can be encrypted in application level will give more secure than other type of encryption such as File level Encryption and Full-Disk encryption. Below table will shows the advantages of application level encryption.

RISK	Full Disk Encryption	File Level Encryption	Application Encryption-Privileged Users
Data unrecoverable when drive stolen or lost from data center	Yes	Yes	Yes
Data made inaccessible to root and system admins	No	Yes	Yes
Data made inaccessible to admins	No	Yes	Yes
Create access logs for threat analytics	No	No	Yes
Unstructured data , config files, logs protected from theft	Yes	Yes	Yes

Table 1: Different level of Encryption along with Risk

In application level encryption we are purposing Key generation & Policy Management, Encryptioning the Sensitive Data, Decrypting the Sensitive Data for authorized users, privileged user access control management

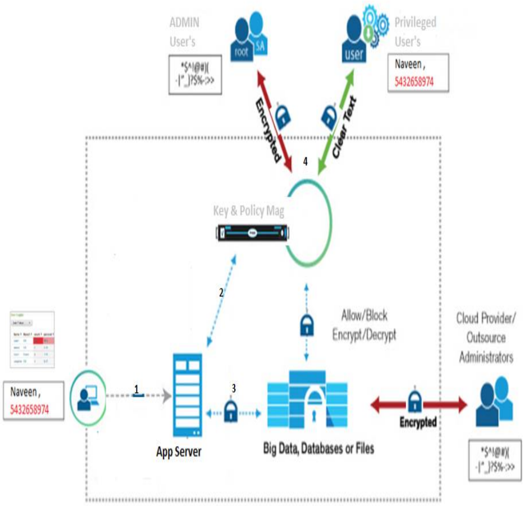


Fig 1: Proposed to cryptosystem along with AppServer , Key & Policy Management

- Key Management & Policy Management:-

In this Key Management phase privileged user's will get the users key, using this key user can encrypt and decrypt the sensitive data. To generating the Key Generation we can use the public key cryptosystem like P^MQ^N-RSA[3][4][5][6] etc., Policy management will classify the sensitive data from the file so sensitive data can't be tampered or hacked from other users such as Admin, Cloud Provider & Outsource Administrators of Cloud.

Sensitive and Non-Sensitive Data is moving to Big Data Cluster's we are following below steps

Key Generation: - Privileged user's creating the keys using the P^MQ^N-RSA[3][4][5][6]which is elaborated in the below

IV. PSEUDO CODE

P^MQ^N-RSA algorithm:

We use the PMQN RSA algorithm as a basis to provide data-centric security for shared data:

```
Step 1 :Randomly chosen distinct primes P, Q
Step 2 :Randomly chose the two natural numbers M,N
Step 3 :Calculate $n = P * Q$
Step 4 :Calculate  $Ø(n) = (P^{M-1} -1)* (Q^{N-1} -1)$
Step 5 :Select e such that e is relatively prime to Ø (n) and less than Ø (n).
Step 6 :Calculate d such that d is e congruent modulo 1 (modØ (n)) and d<Ø (n).
Step 7: Public key = {e, n}
Step 8 :Private key = {d, n}
Step 9 :Cipher message  $c = ( msg^e)mod\ n$
Step10: Decipher msg=  $c^d mod\ n= (( msg^e)mod\ n)^d\ mod\ n = msg^{ed}\ mod\ n = msg$ = Plain msg
```

V. WORKFLOW

Step 1:- User's data having sensitive and non-sensitive data transferring to the App Server's

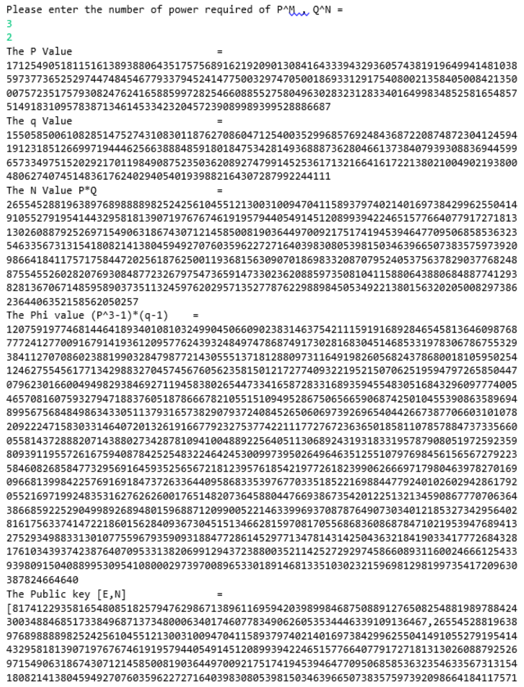
Step 2:- Data is moving\transferring to the Big Data cluster's through App Server, while transferring the data through App Server we need encrypt the sensitive data using the Key & Policy Management. Key Management will generate keys and distributing to the group or user's using the private using P^MQ^N-RSA algorithm as shown above.

Step 3:- Privileged user's Key and Policy classification (HIPAA\ PCI DSS) sensitive data is encrypting and storing in the Big Data clusters

Step 4:- while accessing sensitive data, primarily the system will check user's Key and their policy in Key Management and Policy Management after successful authentication privileged users can decrypt the sensitive data. If non-privileged user's (Admin's, Root user's, Cloud Provider / Outsource Administrators) trying to access the sensitive data they will receive the encrypted data.

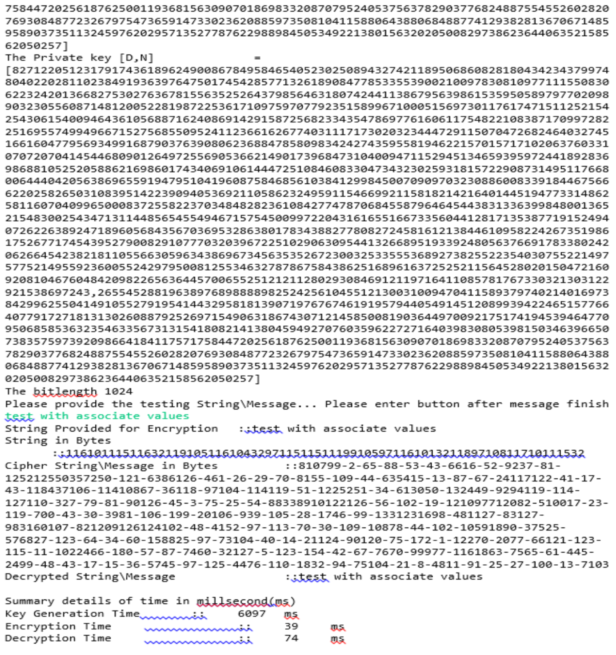
VI. PSEUDO CODE

The simulation studies involve the case study of cryptosystem along with test results, the cryptosystem is deployed using the java language and run with sample of 100 test cases with 1024 bits



```
Please enter the number of power required of P^M , Q^N =
3
2
The P Value                            =
17125490518115161389388064351757568916219209013084164333943293605743819196499414810385
97377365252974474845467793379452414775003297470500186933129175408002135840500842135000
75723517579308247624165885997282546608855275804963028323128333401649983485258165485751
4918310957838713461453342320457239089989399528886687
The q Value                            =
15505850061082851475274310830118762708604712540035299685769248436872208748723041245941
91231851266997194446256638884859180184753428149368887362804661373840793930883694459965
73349751520292170119849087523503620892747991452536171321664161722138021004902193800480
627407451483617624029405401939882164307287992244111
The N Value P*Q                        =
26554528819638976898888982524256104551213003100947041158937974021401697384299625504149
10552791954144329581813907197676746191957944054914512089939422465157766407791727181313
02608879252697154906318674307121458500819036449700921751741945394647709506858536323546
33567313154180821413804594927076035962272716403983080539815034639665073835759739209866
41841175717584472025618762500119368156309070186983320870795240537563782903776824887554
55260282076930848772326797547365914733023620885973508104115880643880684887741293828136
70671485958903735113245976202957135277876229889845053492213801563202050082973862364406
352158562050257
The Phi value (P^3-1)*(q-1)    =
12075919774681446418934010810324990450660902383146375421115919168928465458136460987687
77241277009167914193612095776243932484974786874917302816830451468533197830678675532938
41127070860238819903284798772143055513718128809731164919826056824378680018105950254124
62755456177134298832704574567605623581501217277409322195215070625195947972658504470796
23016600494982938469271194583802654473341658728331689359455483051684329609777400546570
81607593279471883760518786667821055151094952867506566590687425010455390863589694899567
56848498634330511379316573829079372408452650606973926965404426673877066031010782092224
71583033146407201326191667792327537742211177276723636501858110785788473733566005581437
28882071438802734287810941004889225640511306892431931833195787908051972592358093911955
72616759408784252548322464245300997395026496463512551079769845615656727922358460826858
47732956916459352565672181239576185421977261823990626669717980463978270169096681399842
25769169184737263364409586833539767703351852216988447792401026029428617920552169719924
83531627626260017651482073645880447669386735420122513213459086777070636438668592252904
99892689480159688712099005221463399693708787649073034012185327342956402816175633741472
21860156284093673045151346628159708170556868360868784710219539476894132752934988331301
07755967935909318847728614529771347814314250436321841903341777268432817610343937423876
40709533138206991294372388003521142527292974586608931160024666125433939809150408899530
9541080002973970089653301891468133510302321596981298199735417209630387824664640
The Public key [E,N]                   =
[81741229358165480851825794762986713896116959420398998468750889127650825488198978842430
03488468517338496871373480006340174607783490626053534446339109136467,2655452881963897
68988889825242561045512130031009470411589379740214016973842996255041491055279195414432
95818139071976767461919579440549145120899394224651577664077917271813130260887925269715
49063186743071214585008190364497009217517419453946477095068585363235463356731315418082
14138045949270760359622727164039830805398150346396650738357597392098664184117571
```

Fig. 2. Sample Code Result of 1024 bits



```
758447202561876250011936815630907018698332087079524053756378290377682488755455260282076
93084877232679754736591473302362088597350810411588064388068488774129382813670671485958
90373511324597620295713527787622988984505349221380156320205008297386236440635215856205
0257]
The Private key [D,N]                  =
[82712205123179174361896249008678495846540523025089432742118950686082818043423437997480
40220281102384919363976475017454285771326189084778533553900210097830810977111550830622
32420136682753027636781556352526437985646318074244113867956398615359505897977020989032
30556087148120052281987225361710975970779235158996710005156973011761747151125215425430
61540094643610568871624086914291587256823343547869776160611754822108387170997282251695
57499496671527568550952411236616267740311171730203234447291150704726824640327451661604
77956934991687903763908062368847858098342427435955819462215701571710206376033107072070
41454468090126497255690536621490173968473104009471152945134659395972441892836986881052
52058862169860174340691061444725108460833047343230259318157229087314951176680064440420
56386965591947951041960875846856103841299845007090970323088600833918446756662202582650
31083951422390940536921105862324959115466992115818214216401445194773314862581160704099
65000837255822370348482823610842774787068455879646454438313363998480013652154830025434
71311448565455494671575450099722043161655166733560441281713538771915249407262263892471
89605684356703695328638017834388277808272458161213844610958224267351986175267717454395
27900829107770320396722510290630954413266895193392480563766917833802420626645423821811
05566305963438696734563535267230032533555368927382552235403075522149757752149559236005
52429795008125534632787867584386251689616372525211564528020150472160920810467604842098
2265636445700655251212112802930846912119716411085781767330321303122921538697243,2655452
88196389768988889825242561045512130031009470411589379740214016973842996255041491055279
19541443295818139071976767461919579440549145120899394224651577664077917271813130260887
92526971549063186743071214585008190364497009217517419453946477095068585363235463356731
31541808214138045949270760359622727164039830805398150346396650738357597392098664184117
57175844720256187625001193681563090701869833208707952405375637829037768248875545526028
20769308487723267975473659147330236208859735081041158806438806848877412938281367067148
59589037351132459762029571352778762298898450534922138015632020500829738623644063521585
62050257]
The bitlength 1024
Please provide the testing String\Message... Please enter button after message finish
test with associate values
String Provided for Encryption   ::test with associate values
String in Bytes
       ::1161011151163211910511610432971151151119910597116101321189710811710111532
Cipher String\Message in Bytes          ::810799-2-65-88-53-43-6616-52-9237-81-
125212550357250-121-6386126-461-26-29-70-8155-109-44-635415-13-87-67-24117122-41-17-
43-118437106-11410867-36118-97104-114119-51-1225251-34-613050-132449-9294119-114-
127110-327-79-81-90126-45-3-75-25-54-88338910122126-56-102-19-121097712082-510017-23-
119-700-43-30-3981-106-199-20106-939-105-28-1746-99-1331231698-481127-83127-
983160107-821209126124102-48-4152-97-113-70-30-109-10878-44-102-10591890-37525-
576827-123-64-34-60-158825-97-73104-40-14-21124-90120-75-172-1-12270-2077-66121-123-
115-11-1022466-180-57-87-7460-32127-5-123-154-42-67-7670-99977-1161863-7565-61-445-
2499-48-43-17-15-36-5745-97-125-4476-110-1832-94-75104-21-8-4811-91-25-27-100-13-7103
Decrypted String\Message                ::test with associate values

Summary details of time in millsecond(ms)
Key Generation Time      ::     6097   ms
Encryption Time          ::     39     ms
Decryption Time          ::     74     ms
```

Fig. 3. Encryption and Decryption Sample Code Result of 1024 bits

VII. CONCLUSION

In this paper we have implemented P^MQ^N-RSA algorithm for encrypt the sensitive data to the file for privileged user's after applying the policy classification. Using the above model it's hard to hack or tamper the sensitive data for non-privileged user's such user's (Admin's, Root users, Cloud Provider / Outsource Administrators). From the results we obtained it is proved that P^MQ^N-RSA gives more protection only authorized user can retrieve the encrypted data and decrypt it.

REFERENCES

1. Cong Wang, Qian Wang, and Kui Ren and Wenjing Lou (2011) "Ensuring Data Storage Security in Cloud Computing"- IEEE Transaction on Parallel and Distributed Systems, Vol. 22, No. 5
2. Kenneth Cukier, Data, Data Everywhere, Economist, Feb. 27, 2010, at 3-5, available at http://www.economist.com/node/15557443.
3. NavaneetOjha and SahadeoPadhye "Cryptanalysis of Multi Prime RSA with Secret Key Greater than Public Key" International Journal of Network Security, Vol.16, No.1, PP.53-57, Jan. 2014
4. Saveetha&S.Arumugam "Study on improvement in RSA algorithm and its implementation" International Journal of Computer & Communication Technology ISSN (PRINT): 0975 - 7449, Volume-3, Issue-6, 7, 8, 2012.
5. Suganya .N , "Implementing Multiprime RSA Algorithm to Enhance the Data Security in Cloud Computing", International Journal of Innovative Research in Science, Engineering and Technology, ISSN(Online): 2319 - 8753
6. R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signature and Public Key Cryptosystems", Communications of the ACM, Vol. 21, No. 2, pp. 120-126 (1978).
7. Wang Jun-jie&MuSen (2011) "Security Issues and Countermeasures in Cloud Computing" IEEE Conference on Grey Systems and Intelligent Services (GSIS).
8. Wentao Liu (2012) "Research on Cloud Computing Security Problem and strategy" 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)
9. YashpalKadam, "Security Issues in Cloud Computing A Transparent View", International Journal of Computer Science Emerging Technology, Vol-2 No 5 October, 2011 , 316-322.

BIOGRAPHY

Naveen Kumar R isa Research Scholar in the Computer Science Department, SVUCM&CS, SVUniversity Campus, Thirupathi, He received Master of Computer Application (MCA) degree in 2008 from SVUniversiry, Tirupati, AP, India. His research interests are Big Data, Cryptosystems and Computer Networks etc.

Prof. M. Padmavathammaisa BoS chairperson, Dept.of Computer Science, S.V.University, tirupati. She is an eminent professor and awarded nearly 16 Ph.D and 11 M.Phil., under her supervision. She got Best Teachers Award in 2014 from Govt of Andhra Pradesh, India. She delivered various keynote addresses on Network Security & Cryptography at various locations like Kuwait, China, Mauritius , Malaysia, and Singapore .She is an editorial member and reviewer for various well-known journals. Her area of research includes Network Security, Privacy Preserving Data mining, Cloud Computing, Artificial Intelligence and Image Processing.