ISSN 2395-1621

Security and Privacy in Information Brokering System

[#1]Hrushikesh Bhosale, [#2]Amit Chaudhari, [#3]Amar Karande, [#4]Ranjit Phadatare

[1]hrushi555bhosale@gmail.com,
[2]imamitchaudhari@gmail.com,
[3]amarkarande567@gmail.com,
[4]ranjitphadatare781@gmail.com

[#1234]Department of Computer Engineering
JSPM'S
Imperial College of Engineering & Reserch
Wagholi, Pune-412207

ABSTRACT

In public network/database the practical solution to the problem is applied using our scheme. In our system, user forwards the query for requesting data to the broker which then forwards it to the corresponding co-ordinator. Brokers as well as co-ordinator's both are trusted and authorized by central authority (CA). IBS has truthful assumptions on brokers who can fulfil the requirements of user by locating right data provider where necessary data is present. With rising concerns on protecting the sensitive data, the organizations prefer information sharing in a privacy-preserving way, instead of merely full trust on brokers as the brokers may leak information to unauthorized users or even be hacked. It follows that sensitive data needs to be encrypted before outsourcing for data privacy or user privacy. System uses asymmetric key cryptographic algorithm, where user data is encrypted using a public key and decrypted at a user side using private key. The proposed system integrates security enforcement, load balancing and query routing while preserving system-wide privacy.

ARTICLE INFO

Article History

Received: 27th May 2016

Received in revised form :

27th May 2016

Accepted: 29th May 2016

Published online :

30th May 2016

Keywords: **Security, Privacy, Information sharing, Load balancing, Access control.**

I. INTRODUCTION

In order to assure security and privacy of users sensitive data stored in a data providers repository, a commonly adopted approach is to encrypt the data before forwarding it to the user. Since the system uses public key for encryption and private key for decryption, the confidentiality of the user's data is assured. Public key cryptography finds its strongest application when parties do not have any former relationship wants to exchange sensitive data with each other. RSA implements a public key cryptography that allows secure communication and digital signatures, and its security rests in part on the difficulty of factoring large numbers. Despite hundreds of years of study of the problem, finding the factors of a large number still takes long time in general. The fastest current methods are much faster than the simple way of trying all possible factors one at a time. However, they are still costly. For instance, it has been estimated recently that finding the prime factors of a 1024-bit number would take a year on a machine. A 2048-bit number would require quite a few billion times more work. However a traditional 1024-bit key RSA has several issues that could potentially damage RSA's security, such as timing attacks and problems with key distribution. In fact these issues have solutions; the only downside is that any device implementing RSA would have to have much more hardware and software to counter certain types of attacks or attempts at eavesdropping. To address such shortcoming, we introduce 2048 bit key which overcomes the limitations of earlier 1024 bit length key. 1024 bit key is faster than 2048 bit key but experts predict that it may be breakable in the near future, so we need stronger key than earlier version. 2048 bit key results in much more security and performance.

II. RELATED WORK

A) RSA ALGORITHM

We introduce the idea of asymmetric encryption, in which a key required to encrypt data is made public, but the corresponding key required to decrypt is kept private, for example in a file on the server to which clients connect. Such a system solves the problem of how to send a temporary encryption key securely to the server when opening a secure connection. A widespread asymmetric encryption system is

© 2015, IERJ All Rights Reserved

RSA, named after inventors Rivest, Shamir & Adleman. RSA encryption and decryption are basically mathematical operations. As one-off process, we need to generate an RSA key pair that from then on, we'll use for all conversations between our clients and server. Creating an RSA key pair basically consist of getting a modulus, which is based on two random primes proposed to be unique to that key pair, picking a public exponent then calculating the equivalent private exponent given the modulus and public exponent. In fact, we need to store the public and private keys someplace. Usually, the private key will be placed on our server, and the public key distributed to clients. To store the key, we just need to pull out the modulus and the public-private exponents, then write these numbers to some file or put in whatever well-situated place. In 800-57, NIST advises that 1024-bit RSA keys will no longer be feasible and advises to move to 2048-bit RSA keys. NIST advises that 2048-bit keys should be feasible until 2030. Some people have preference on RSA 4096-bit keys, considering "longer is better". However, "longer is better" is not always true. When it's long, it requires more computational resources, memory and storage, and it consumes more power for normal usages. These days, many people have enough computational resource, that would be true, but less is better for power consumption. For security, the key length is just a single factor. We had and will have algorithm issues, too. It is true that it's difficult to update our public keys, but this problem wouldn't be solved by just having longer keys.

III. IMPLEMENTATION

Key Generation

1) Pick two large prime no's, p & q, p!=q.
2) Calculate n=p*q
3) Calculate m or ϕ(n)=(p-1)*(q-1)
4) Pick e, so that GCD(e, ϕ(n))=1, 1<e<ϕ(n)
5) Calculate d, so that d*e mod ϕ(n)=1, i.e, d is multiplicative inverse of e in mod ϕ(n)
6) Get Public key as K_U={e, n}
7) Get Private key as K_R={d, n}
8) Encryption, $C=P^e \bmod n$
9) Decryption, $P=C^d \bmod n$

IV. EXISTING SYSTEM

The existing system PPIB (Privacy Preserving Information Brokering) scheme is for secure data transmission. Although this scheme is primarily based on RSA-1024 bit key encryption algorithm which is a asymmetric key encryption scheme for secure data transmission with the concept of public-private key. Basically it is a great solution for different attacks like Attribute-correlation attack and inference attack. Despite recent advancements in implementation techniques, currently this encryption scheme is prone to different kinds of attacks as it uses only 1024 bits of key for encryption and decryption so it seems to be less secure due to recent advancements in factorization algorithms.

V. PROPOSED SYSTEM

Now a day there is an increasing need for inter-organizational information sharing to facilitate extensive collaboration. At the same time it is challenging task to handle heterogeneous data and provide interoperability for the same. In many applications we need enforcement information sharing, in which organizations share information in a conservative and controlled manner due to business considerations or legal reasons. Here we are providing new emerging technique for hospital information systems. Our aim is to facilitate access to and retrieval of hospital data across collaborative healthcare providers that include a number of regional hospitals, on-line patients requirement etc. The privacy of a data owner (Hospital) is the identifiable. Data and sensitive or personal information carried by this data (Hospital patient's records). Data providers are authorized and trusted to prevent unauthorized use or disclosure. Data providers store the collected data locally and create two types of meta-data, namely routing meta-data and access control meta-data, for data brokering. Both types of meta-data are considered privacy of a data provider. Data requesters may reveal identifiable or private information (e.g., information specifying his interest) in the querying content. For example, a query about USER reveals the user details. Using RSA with 2048 bit length key we can defeat different kinds of attacks like attribute-correlation attack, inference attack, etc. XML file for users information will get created here. Emergency access is also provided to authorized doctors.

VI. SYSTEM ARCHITECTURE

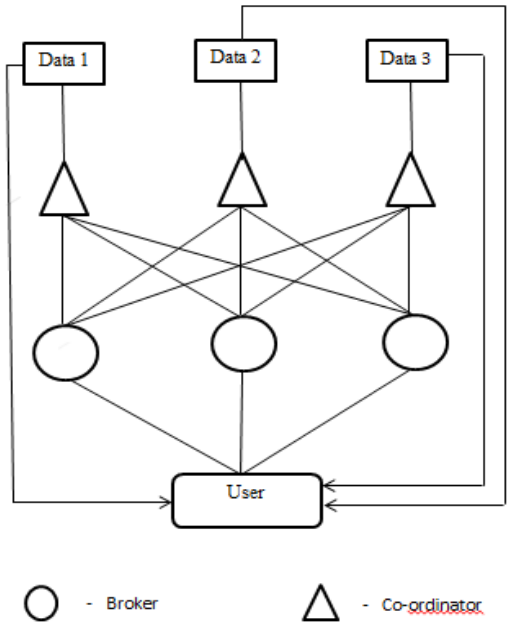

Fig 1. Architecture of SPIBS.

User sends query to broker for requesting data. Broker who is free to handle query responds to the users query. Broker then forwards it to the particular Coordinator who in turns sends the requesting data directly to the user. Here CA (Central Authority) performs the task of authorization. Both brokers and coordinators are authorized by CA. Brokers are the main entities which performs load balancing and query routing.CA allocates different work for each coordinator who in turns performs its assigned task. In case, if any broker leaves the system or work then CA has pending request of brokers who are interested to join. CA can appoint any of the brokers from

© 2015, IERJ All Rights Reserved

requesting brokers queue. In case of emergency doctors has authority to access reports of the user.

VII. MODULES

Coordinator Module It is responsible for content-based query routing and access control. Each coordinator holds segment of access control and routing guidelines. It receives query from broker and forwards the requested data to the intended user.

Broker Module It is mainly responsible for user authentication and query routing. It acts as a mediator between coordinator and data user. The request received by broker is verified and passed to the coordinator.

User Module It forwards the query for requesting data to the broker.

CA (Central Authority) Module Perform the task of authorization. Brokers and coordinators are selected by CA. It assigns job to each coordinator.

VIII. MATH OF THE MODEL

Encryption, $C = E(M) = M^e \bmod n$.

Decryption, $D = D(C) = C^d \bmod n$.

IX. RESULT AND DISCUSSION

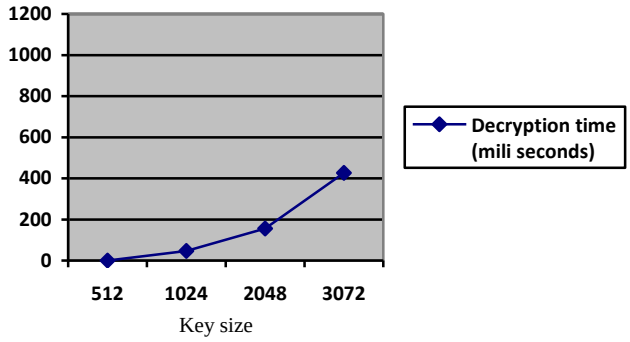

Figure No.2

Above figure no.1 shows the decryption time taken by keys of different length. Short keys unusually require less decryption time but they are less secure.

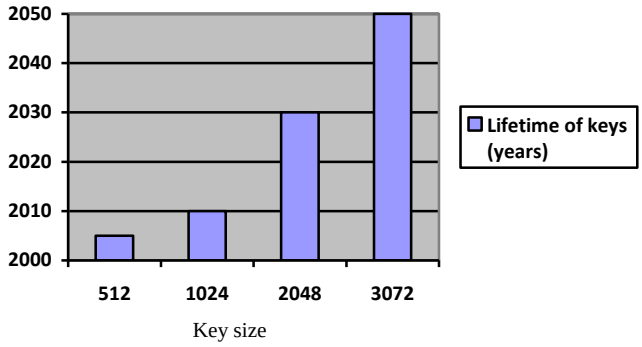

Figure No.2

Here Figure No. 2 shows the predicted lifetime of each key, by NIST. 1024 bit key was secure till 2010. NIST says that 2048 bit RSA keys will be secure till 2030. After that more secure 3072 bit key can be used. But we have to understand that more the key size more the decryption time taken by algorithm.

Here RSA algorithm provides security and confidentiality to user's sensitive data with public-private key schema. System provides the reasonable security to the user's private data and it operates fast enough as compared to the hardware used. The entire process is easy so that anyone can use it efficiently.

X. CONCLUSION

We invent and undertake important problem of security of user's sensitive data in hospitals. Security and Privacy in Information Brokering System uses RSA algorithm to provide security to users sensitive data like CTR, PTR, BTR, etc. reports has been discussed in this paper. As per our study and discussion, RSA algorithm provides efficient security and confidentiality to sensitive data and it is one of the best asymmetric key cryptographic algorithm based on factoring of large prime no's. The implemented system focuses on avoiding leakage of sensitive data to unauthorised person. General research has been conducted to verify the effectiveness and efficiency of our proposed approach.

ACKNOWLEDGMENT

The authors wishes to thank Prof. Vinod Wadne(Guide), Prof. Darshika Lothe(PG-Coordinator), Prof. S. R. Todmal(HOD) and Dr. Sachin Admane(Principal) for valuable guidance and encouragement.

REFERENCES

[1] A. Fengjun Li, Bo Luo, Peng Liu, Dongwon Lee, and Chao-Hsien Chu "Enforcing Secure and Privacy-Preserving Information Brokering in Distributed Information Sharing" IEEE Trans. Information Forensics and Security, Vol. 8, No. 6, June 2013.

[2] Himanshu Thapliyal and M.B Srinivas, "VLSI Implementation of RSA Encryption System Using Ancient Indian Vedic Mathematics",

[3] L. M.Haas, E. T. Lin, andM.A. Roth, "Data integration through database federation," IBM Syst. J., vol. 41, no. 4, pp. 578–596, 2002.

[4] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "CoolStreaming/DONet:A data-driven overlay network for efficient live media streaming," in Proc. IEEE INFOCOM,Miami, FL, USA, 2005, vol. 3, pp. 2102–2111.

[5] A. C. Snoeren, K. Conley, and D. K. Gifford, "Mesh-based content routing using XML," in Proc. SOSP, 2001, pp. 160–173.

[6] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes," in *Proc. Crypto '98*, H. Krawczyk Ed. Springer-Verlag, LNCS 1462.

© 2015, IERJ All Rights Reserved

[7] Suchita Tayde," File Encryption, Decryption Using AES Algorithm in Android Phone," Volume 5, Issue 5, May 2015.

[8] N. Koudas, M. Rabinovich, D. Srivastava, and T. Yu, "Routing XML queries," in Proc. ICDE'04, 2004, p. 844.

[9] G. Koloniari and E. Pitoura, "Peer-to-peer management of XML data: Issues and research challenges," SIGMOD Rec., vol. 34, no. 2, pp.6–17, 2005.

[10] F. Li, B. Luo, P. Liu, D. Lee, and C.-H. Chu, "Automaton segmentation: A new approach to preserve privacy in XML information brokering," in ACM CCS '07, pp. 508–518, 2007.

[11] Bertino, S. Castano, and E. Ferrari, "Securing XML documents with author-x," *IEEE Internet Comput.*, vol. 5, no. 3, pp. 21–31, May/Jun. 2001.

© 2015, IERJ All Rights Reserved