Journal of Theoretical and Applied Information Technology
31st August 2016. Vol.90. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PROVIDING SECURITY TO PUBLISH / SUBSCRIBE SYSTEM USING STRONG NETWORK CRYPTOGRAPHIC TECHNIQUES AND TWO TIER ENCRYPTION

[1]AANCHAL RAI, [2]MRS. MAYA SHELKE - BEMBDE
[1]Student: M. Tech (CSE) Symbiosis Institute of Technology Pune, Maharashtra, India
[2]Asst. Prof. CS & IT Dept. Symbiosis Institute of Technology Pune, Maharashtra, India

E-mail: aanchal.rai@sitpune.edu.in, mayas@sitpune.edu.in

ABSTRACT

There are different types of distributed systems existing in this world which needs the massive amount of information to be transferred between them and along with it comes the need of secure communication between different parties. Publish subscribe system is the communication archetype for information broadcasting in huge distributed system. Selling a product and merchandising it, isn't a cup of tea for each manufactures, thus all manufactures are probing for various third party product promoters like promotion agencies, dealers or third party brokers. For a business to explore, brokers are playing a very crucial role. However loyalty of the brokers cannot be assured. So to handle this problem many broker less systems are been proposed for publish-subscribe system in distributed pattern. However most of these systems have not completely succeeded in achieving privacy of the data over the transactions. So proposed system forwards an idea of preserving the confidentiality of information by key generation for each publish/ subscribe transactions and the keys are randomly generated based on the event details and subscriber data using reverse circle cipher[10] and two tier encryption.
Keywords: *Cryptography, Publish-Subscribe, Broker-Less, Two-Tier Key, Gaussian Distribution*

1. INTRODUCTION

Publish subscribe is one of the most prominent method for communication in distributed systems. It is one amongst the best effective methodology where group communication is there. Publish subscribe [2], [12], [20] is communication archetype for distributed systems. It includes publisher who publishes the information or the one who send the messages or data and subscriber who subscribes or who receives the data sent by the publisher. Publisher publishes the information and subscriber's shows their interest to only those information which they want to receive so subscriber can then receive solely those data within which they're interested. There are variety of applications [3] existing which uses this system specially when there is the need of cluster communication. Publish subscribe system is employed broadly speaking in embedded system wherever each component respond during a real time for events during a list. The main reason because of which pub sub system is used so much is its decoupling property. Publisher and subscriber are not tightly bounded with each other instead they are loosely coupled and thus it acquires scalability property in word of number of sender and receiver. Initially research in pub sub system is usually related to scalability but then then it comes to security. Here security [1], [7] is related to security of data transmission between publisher and subscriber confidentiality of sender and receiver as well as confidentiality between different subscribers. In publish subscribe system achieving security is quite tough task because of loose coupling. Hence access control concept came into the picture. Access control [3], [5] in relation with pub sub system means that only authenticated publisher can send the message and it is received only by authorized subscriber. Publisher and subscriber don't believe on one another on trust issue. Usually all of them are honest but still there may be chance of having fake publisher who tries to fool others by pretending to be real publisher and similarly there can be fake subscribers who will fool and get the data which they are not authorized to receive.

This system provides the confidentiality [6], [15] for data of the Event owner in content based

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

publish subscribe system [8], [11], [17], [18], [19]. Only authorized subscriber can access the event data. Event data will be always stored in encrypted format at publishers end. Publisher re-encrypts the data using two tier key system for more security [1], [7]. On accessing data from the publisher subscriber gets the decrypted event data.

The following are the main features that are included in the broker-less system [9]:

1. Publisher- who generate the event data

2. Broker- publish the event

3. Subscriber- subscribes the event and access the event

The system will perform the following functions: It should encrypt [13], [14], [16] the sender's data and store in the file format on event creation; it should decrypt the file when accessed by the subscriber; it should create key based on the profile; it should create key based on the time; it should distribute the event based on the subscribers request pool.

The remaining paper is arranged as follows: Section II demonstrates the literature survey of the different techniques for providing security and different applications using publish subscribe techniques. Section III describes the proposed methodology which includes architecture and the different algorithms for encryption and key generation. Section IV gives results and graphs. Section V gives conclusion and future scope.

2. LITERATURE REVIEW

Various approaches have been proposed for providing security in publish-subscribe system:

Mudhakar Srivatsa, et.al [1] have proposed Event Guard, a reliable design that protects the publisher subscriber work from various types of attack. It provides security characteristics that are critical to publisher and subscriber overlay network benefits, like integrity, confidentiality, authenticity, providing flexibility to continuous flood based denial of service attack.

Jean Bacon David, et.al [2] tells the needs of numerous application areas where the event-based design is suitable but security is matter of concern. They have discussed about how security can be provided to publish subscribe system; first they tell about the access control policy at service API and then apply it, and they then apply the various security form of these policies in the service network.

John Bethencourt , et.al [3] presented a system for achieving access management on cipher text that is known as Cipher text-Policy Attribute-Based Encryption.[3] Through this techniques confidentiality of data is ensured also if storage server is untrustable.

Vipul Goyal, et.al [4] proposed a cryptographic system for fine-grained sharing of cipher text which is called as Key-Policy Attribute-Based Encryption (KP-ABE) [4]. In this system, encrypted data are labeled with some collection of attributes and along with it the private keys have their relation with access structures which can be decrypted using the cipher text. It additionally tells the relevance of construction to sharing of audit-log data and broadcast coding. The development supports devolution of personal keys which incorporates Hierarchical Identity-Based Encryption (HIBE)[9].

Mihaela Ion, et.al [5] presented a peculiar system which supports confidentiality for events as well as filters; filters apply complex restraint on events although brokers cannot access data through filters as well as events and therefore there is no need of sharing of keys between publisher and subscriber.

Himanshu Khurana, et.al [6] narrates the idea of flexible solutions for security for publisher-subscriber systems which includes confidentiality, authenticity and integrity of the system. They have provided the usage based accounting services [6], that is employed by e-commerce applications which uses pub/sub systems. This kind of solution is suitable in a scenario where sender and receiver both don't trust the pub/sub environment.

Costin Raiciu, et.al contributions involves an analysis of the problem, which shows that attaining maximum level of security in our scenarios is quite restricted, and even obtaining security for small level is also costly for generic subscription functions. They have focused on providing confidentiality for commonly used applications and subscription in a content-based publish/subscribe system[7] and they have also

Journal of Theoretical and Applied Information Technology
31st August 2016. Vol.90. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

proposed a series of practical solutions that they have embedded in the implementation of Siena, a popular content-based pub/sub system[7].

Muhammad Adnan Tariq, et.al [9] contribution involves the utilization of searchable encryption for valuable routing of cipher text event [9] and proposed the multicredential routing, event dissipation action to empower the weak subscription confidentiality, and they have also done the analysis of different type of attacks to check the confidentiality between subscribers. The whole idea gives the fine-grained key management [9].

Ebenezer R.H.P. Isaac, et.al have proposed the Reverse Circle Cipher [10] algorithm for encryption and decryption which is better than other systems in terms of time and space complexity. This encryption technique is used in standalone structures for private information confidentiality and also used in network security in case of real time scenario [10]. They have done the study on how effective reverse circle cipher[10] algorithm in terms of size on plain text and frequency distribution on cipher text[10].

Choi[11], et.al presents the approach which solves one of the major problem of broker less publish subscribe system. Since in this the secure contents cannot be revealed to broker as broker are not trustable. Because of this broker face some problems in doing certain operations. There are some techniques which attempted to solve this problem and it has even succeeded, but its drawback is that it is very expensive. So to solve this problem, this paper proposes the new method for dissemination of data which is called as Asymmetric Scalar product preserving encryption (ASPE).

FengTian[20], et.al presents the approach for improving the performance of XML publish subscribe system. For XML pub sub system, there is a matching of subscription i.e. queries against documents of the XML. This takes more time when numbers of queries are in large number. Some approaches tries to solve this problem. A finite automaton was made up for storing the queries in memory space. So this paper proposes the system where relational database is used for matching purpose. System has implemented and proofs that this approach gives the better performance and is very scalable approach for large number of subscriptions.

3. PROPOSED IDEA

The system proposes an idea of random key generation for each publish/ subscribe transactions which is generated based on the details of event publishers profile. Gaussian distribution model acts as catalyst in this whole process to improve the privacy of the system blended with fine grained cryptographic models. The proposed idea comes by the fact that random keys can be maintained by the random request parameters done by the subscriber. Then to improve the complex key structure, system uses Gaussian distribution model expeditiously together with extremely highly secured Reverse circle cipher for maintaining privacy. To enforce the system more robustly system uses high level key management system within the network.

It is a distributed system implementing in client-server model. The broker-less system provides easy mechanism for the event publisher and subscribers to meet their need of event knowledge.

3.1 Proposed Architecture

In this section, we describe our framework for broker less publisher / subscriber system [9] using strong network cipher
techniques with the below mentioned steps as shown in figure

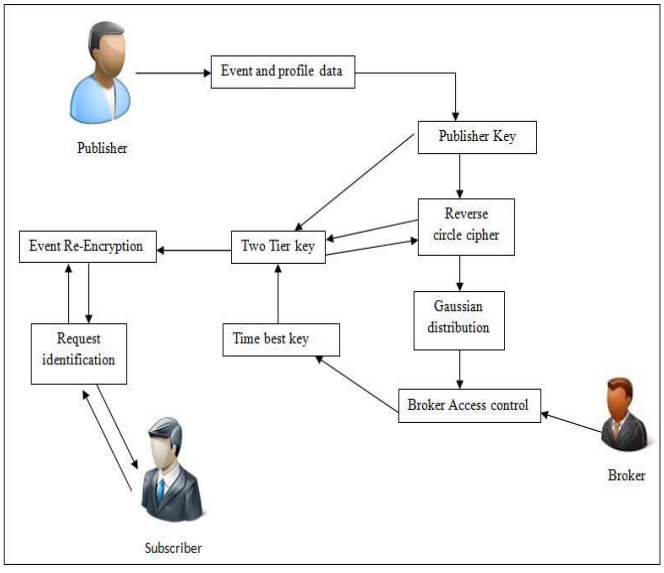

Fig 1: Overview of our approach

Journal of Theoretical and Applied Information Technology
31st August 2016. Vol.90. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 1: This is the most primitive step where all the event data and profile data of the owner is been concatenated to create a profile key using the random key generation algorithm as mentioned in the step 2.

Step 2: Here Random private key is generated using the random key generation for the users profile data and this is act as a private for each user transactions. This is shown as below.

ALGORITHM 1: RANDOM KEY GENERATION

Input: Instance Date and time in String
Output: Key
Step 0: **Start**
Step 1: Get the instance time and date in String Called "Ds"
Step 2: Remove Special Symbols from $\mathbf{D_s}$ (Like /, -)
Step 3: Get the **MD5** Hash key of $\mathbf{D_s}$ in String as **H**
Step 4: Assign sum=0, **Key** ="”
Step 5: **for** i=0 to length of $\mathbf{D_s}$
Step 6: sum =sum+ASCII of $D_s[i]$
Step 7: End **For**
Step 8: Random integer **R**=sum **MOD** 7
Step 9: **while Key** length is less than 7
Step 10: Select Random character from **H on** index R
Step 11: and concatenate to key
Step 12: rotate **H** by one character
Step 13: End **While**
Step 14: return **Ke**
Step 15**: Stop**

Step 3: Proposed system makes use of reverse circle cipher [10] an encryption algorithm for imposing the strong security policy. Reverse circle cipher is secured compared to other because it makes use of private key for encryption purpose. Once the input string is obtained it is divided into blocks of 10 characters. Then these individual blocks are rotated by their respective index and then fed to the encryption module. Encryption module accepts the rotated string and based on the ASCII value of each of the character encryption is performed. Detail implementation procedure for reverse circle cipher algorithm is explained in below algorithm.

ALGORITHM 2: REVERSE CIRCLE CIPHER

Input: File Text **T** and Key **K**
Output: Encrypted Text TE
Step 0: **Start**
Step 1: Create a vector called **DIV** and initialize count=0, initialize String B to empty
Step 2: **FOR** i=0 to length of T
Step 3: Keep joining characters from **T** into String **B**, and count++
Step 4: **If** count =10
Step 5: Add **B** to **DIV**, set count=0 and empty **B**
Step 6: End **FOR**
Step 7: **FOR** i=0 to Size of **DIV**
Step 8: String B_s= DIV[i]
Step 9: rotate B_s by one character, initialize sum =0
Step 10: **FOR** j=0 to length of **K**
Step 11: sum =sum+ASCII of K[j]
Step 12: END **FOR**
Step 13: Val=sum%20
Step 14: **FOR** j=0 to length of B_s
Step 15: ASCII of $B_s[j]$ + Val
Step 16: Replace a new character
Step 17: **End FOR**
Step 18: Concatenate B_s to a string T_E
Step 19: return T_E
Step 19: **End FOR**
Step 20: **Stop**

Step 4: Here in this step event data is been distributed to different Broker based on the Gaussian distribution model (GDM). Where it considers the distribution parameter as the number of the distributed events by the Broker. Where GDM is a continuous function which approximate the exact binomial distribution of the events by the Broker to give him the right weight.

Gaussian Distribution Equation

$$P(y)=\frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(y-\mu)2}{2\sigma^2}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Where
μ= mean of distribution
σ^2=variance of distribution
y= continuous variable
P(y) = probability of y
Step 5: Here in this step Broker for whom the event is been assigned by the event owner is

Journal of Theoretical and Applied Information Technology
31st August 2016. Vol.90. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

access the event data. And create a two tier key which is empowered with time based key along with owner key. This random key generation is been powered with MD5 one way hashing algorithm.

By using this new key encrypted data is been re-encrypted again, which is been controlled by the both owner and broker.

Step5: Here the published data by the broker can be view by the subscribers and then request for the same to the broker. Then this data with the new two tier key is been served to the subscriber, which eventually decrypt using reverse circle cipher decryption technique to deliver plain text event data to the subscriber.

4. RESULTS & DISCUSSION

The proposed model is developed on java based windows machines which uses Apache tomcat as the server and NetBeans as IDE. For the experiments and evaluation WAPT 8.0 tool is been used to measure the performance of the system for different number of users. And then system is been put into several tests to evaluate it more strictly as mentioned below tests.

4.1 Key Space Complexity and Request processing time

In any system where random keys are been generating are especially under the lenses for their space complexity. Here in our case Key space is managed efficiently as they are directly proportional to the number of keys that are been generated as shown in the figure 1. That is eventually a good sign for the key space complexity. And also system indicates the time required for request processing using WAPT 8.0 tool is also mentioned in the below table.

Table 1: Key space and Request processing time

No of Users	Key Space	Request processing Time
25	350	85
50	700	150
75	1050	224
100	1400	314
125	1750	395
150	2100	485

Table 1: Key space and Request processing time

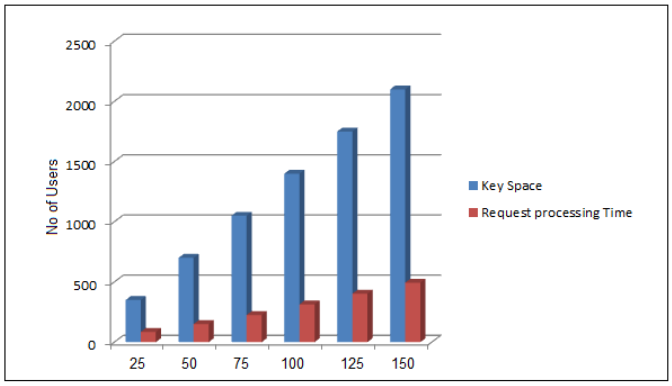

Figure 1: Key Space Complexity analysis

4.2 Character assignment for Encryption

The graph in figure 2 is drawn between the number of file character that are being used for the encryption and decryption v/s number of different characters that are using by the algorithm.

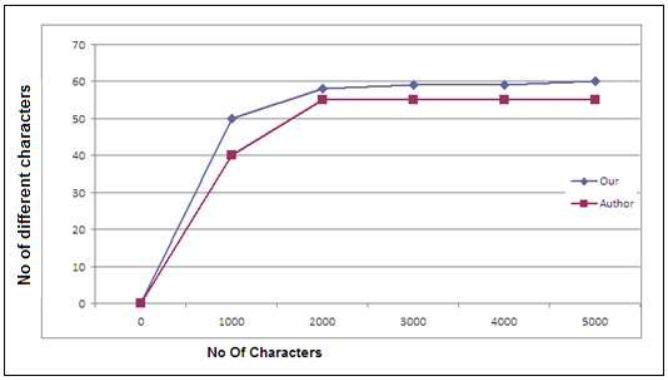

Fig 2: No of File character v/s No of Using different characters for the encryption and decryption

Here in the above graph proposed system of brokering system in web uses more character to encrypt then the system that is been proposed in [10]. This enhances the encryption process to its best and gives fine results.

5. CONCLUSION & FUTURE SCOPE

Proposed method efficiently shows the broker less subscriber / broker relationship without adding much hazards of trustworthiness. Here keys are been generated by permutation of the characters in run time based on the event publisher data generation scenario and broker

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

access scenario with different keys. In this system, publisher is efficiently generating the key based on his profile data and event data. Whereas the broker manages to re-encrypt the data by generating two tier key using publisher key and time based key for the reverse circle cipher encryption cipher base. Again system successfully maintains the Event distribution scenario by using Gaussian distribution model for the broker. And in the end the whole system is tightly coupled to handle many subscriber requests in run time with proper event publishing schemes.

The proposed system can be enhanced to implement in heterogeneous network of internet of things using cluster based hierarchy. This makes easy access of system in all possible types of network.

Cluster based hierarchy in distributed paradigm is the scenario where many clustered node in the systems are assigned for the different work in the distributed network. So we can enhance our model by assigning clusters for handling broker work and event publisher work. This actually greatly reduces the task completion time.

REFERENCES

[1] Stallings, William. *Cryptography and Network Security, 4/E*. Pearson Education India, 2006.

[2] Srivatsa, Mudhakar, Ling Liu, and Arun Iyengar. "Eventguard: A system architecture for securing publish-subscribe networks." *ACM Transactions on Computer Systems (TOCS)* 29.4 (2011): 10.

[3] Bacon, Jean, et al. "Access control in publish/subscribe systems." *Proceedings of the second international conference on Distributed event-based systems*. ACM, 2008.

[4] Bethencourt, John, Amit Sahai, and Brent Waters. "Ciphertext-policy attribute-based encryption." *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE, 2007.

[5] Goyal, Vipul, et al. "Attribute-based encryption for fine-grained access control of encrypted data." *Proceedings of the 13th ACM conference on Computer and communications security*. Acm, 2006.

[6]Ion, Mihaela, Giovanni Russello, and Bruno Crispo. "Supporting publication and subscription confidentiality in pub/sub networks." *Security and Privacy in Communication Networks*. Springer Berlin Heidelberg, 2010. 272-289.

[7] Khurana, Himanshu. "Scalable security and accounting services for content-based publish/subscribe systems." *Proceedings of the 2005 ACM symposium on Applied computing*. ACM, 2005.

[8] Tariq, Muhammad Adnan, et al. "Providing basic security mechanisms in broker-less publish/subscribe systems." *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems*. ACM, 2010.

[9] Tariq, Muhammad Adnan, Boris Koldehofe, and Kurt Rothermel. "Securing broker-less publish/subscribe systems using identity-based encryption." *Parallel and Distributed Systems, IEEE Transactions on* 25.2 (2014): 518-528.

[10] Isaac, Ebenezer RHP, Joseph HR Isaac, and J. Visumathi. "Reverse Circle Cipher for personal and network security." *Information Communication and Embedded Systems (ICICES), 2013 International Conference on*. IEEE, 2013.

[11] Choi, Sunoh, Gabriel Ghinita, and Elisa Bertino. "A privacy-enhancing content-based publish/subscribe system using scalar product preserving transformations." *Database and Expert Systems Applications*. Springer Berlin Heidelberg, 2010.

[12] Jacobsen, Hans-Arno, et al. "The PADRES Publish/Subscribe System." *Principles and Applications of Distributed Event-Based Systems* 164 (2010): 205.

[13] Boneh, Dan, et al. "Public key encryption with keyword search." *Advances in Cryptology-Eurocrypt 2004*. Springer Berlin Heidelberg, 2004.

[14] Boneh, Dan, and Matt Franklin. "Identity-based encryption from the Weil pairing." *Advances in Cryptology—CRYPTO 2001*. Springer Berlin Heidelberg, 2001.

[15] Ion, Mihaela, Giovanni Russello, and Bruno Crispo. "Supporting publication and subscription confidentiality in pub/sub networks." *Security and Privacy in Communication Networks*. Springer Berlin Heidelberg, 2010. 272-289.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[16] Canetti, Ran, Shai Halevi, and Jonathan Katz. "A forward-secure public-key encryption scheme." *Advances in Cryptology—Eurocrypt 2003*. Springer Berlin Heidelberg, 2003. 255-271.

[17] Shikfa, Abdullatif, Melek Önen, and Refik Molva. "Privacy-preserving content-based publish/subscribe networks." *Emerging challenges for security, privacy and trust*. Springer Berlin Heidelberg, 2009. 270-282.

[18] Li, Jun, Chenghuai Lu, and Weidong Shi. "An efficient scheme for preserving confidentiality in content-based publish-subscribe systems." (2004).

[19] Banavar, Guruduth, et al. "An efficient multicast protocol for content-based publish-subscribe systems." *Distributed Computing Systems, 1999. Proceedings. 19th IEEE International Conference on*. IEEE, 1999.

[20] Tian, Feng, et al. "Implementing a scalable XML publish/subscribe system using relational database systems." *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM, 2004.

[21] Montida Pattaranantakul, AroonJanthong, KittichaiSanguannam , ParaminSangwongngam. "Secure and Efficient Key Management Technique in Quantum Cryptography Network" *.2012 Fourth International Conference on Ubiquitous and Future Networks (ICUFN)*

[22] M.RamyaPriyadharshini, S.Prasanna, Dr.N.Balaji. "Energy and Mobility Based Group Key Management in Mobile Ad Hoc Networks". *Recent Trends in Information Technology (ICRTIT), 2014 International Conference*

[23] Ran Canetti,Shai Halevi, Jonathan Kat. "A Forward-Secure Public-Key Encryption Scheme". E. Biham (Ed.): EUROCRYPT 2003, LNCS 2656, pp. 255–271, 2003.International Association for Cryptologic Research 2003.