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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

Privacy Preserving Content Based Publisher/Subscriber System With Auditing and Event based Encryption

Handore Jayshree Shrikant, Shivaji R. Lahane

PG Student, Dept. of Computer Engineering, Affiliated to Savitribai Phule Pune University, GES's R. H. Sapat College of Engineering, Management Studies & Research, Nashik, India

Assistant Professor, Dept. of Computer Engineering, Affiliated to Savitribai Phule Pune University, GES's R. H. Sapat College of Engineering, Management Studies & Research, Nashik, India

ABSTRACT: Content-based publisher/subscriber system has been used by many applications to deliver data among different distributed users. Publish–subscribe system is a message delivery system where messages are send by the publishers, who are not program the messages to sent directly to the receivers also called as subscribers according to their interests. Due to this loose coupling, providing basic security mechanisms like authentication and confidentiality is challenging task in a content-based publish/subscribe system. The Existing broker architecture of most messaging systems have messaging server i.e. broker in the middle similar to star or hub. Every application communicated with the central broker. Applications do not communicate with each other directly. Broker is responsible for all types of communication whether receiving or forwarding of an event to the particular subscriber. As all the communication is done through the broker it requires excessive amount of network communication and as all the messages have to be passed through the broker it becomes bottleneck of the whole system. To address this issue an approach is proposed to provide confidentiality and authentication in a broker-less content-based publish/subscribe system by using the Identity based encryption mechanisms.

KEYWORDS: Content-based, Publish/Subscribe, Peer-to-Peer, Broker-less, Identity Based Encryption

I. INTRODUCTION

As the use of Internet is increasing day by day, it becomes essential to provide more attention towards the security of our data that we are sharing over the Internet. Because there is a possibility that the data is being misused over the Internet by the user who is not authorized to access that data by imper-sonating as an authorized user and have all the permissions that are needed to access that data. This unauthorized access to the confidential data is more harmful in the case where that data is being used for the illegal operations. There is another possibility that passive attacker outside the overlay network can eavesdrop the communication and try to discover content of events and subscriptions. So, to protect the data from unwanted actions of the unauthorized users such as unauthorized access, modification of data etc., there is a need to provide authentication of users, confidentiality of data in order to provide a security to the data. In the traditional communication systems all the communication is based on the request reply communication where any client send request to the server for particular service and after the reply from the server the client can access that particular service. In such a communication systems, there is synchronous, tightly-coupled request invocation so that they are very restrictive for distributed applications, especially for WAN and mobile environments. So, there is a requirement for a more flexible and decoupled communication style that offers anonymous and asynchronous mechanisms.

Publish/subscribe system is a message passing paradigm where the entities who are responsible for the publication of messages are called as publishers; they do not send the messages directly to the receivers of the messages, called as subscribers. Published messages are divided into different classes, without knowledge of number of subscribers in the system. Similarly, subscribers specify their interested classes, and receive messages based on their interest, without knowledge of number of publishers in the system. In pub/sub systems, publishers pass the messages to an intermediate

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

broker and subscribers give their subscriptions to that broker. The broker filters and routes the events from publishers to interested subscribers. The broker normally performs a store and forward function to route messages from publishers to subscribers. There are several advantages to this centralized broker architecture. First advantage is that applications do not have to remember the locations of other applications. They only need network address of the broker. Messages are forwarded to the particular subscriber by the Broker based on their criteria. Second is, message sender and message receiver lifetimes don't have to overlap. Sender application sends the messages to the broker and after that it terminates. Then the receiver application can receive those available messages any time later. Along with these advantages it also brings some disadvantages that are: 1) it requires excessive amount of network communication. 2) As all the messages have to be passed through the broker, it can result in broker turning out to be the bottleneck of the whole system. It becomes a single point of failure. To address this issue a broker-less pub/sub system is proposed in which each node can be publisher, subscriber or broker. Subscribers give their subscriptions directly to the publishers and publishers inform subscribers directly. Therefore they must have knowledge of each other. In the publisher/subscriber system, subscribers receive only those messages to which they have subscribed. The pub/sub systems must provide some sort of access control so that only authenticated publishers are allowed to disseminate events in the network and only those events are delivered to authorized subscribers and the content of events should not be reveal to the routing infrastructure and a subscriber should receive all relevant events without revealing its subscription to the system. In a system, there is a need to allow publishers to sign and encrypt events to maintain the confidentiality of the events, enable efficient routing of that encrypted events (from publishers to subscribers) and also allow the subscribers to verify the signatures associated with all the attributes (of an event). To provide security in broker-less pub/sub systems, this project presents a new approach to provide authentication and confidentiality in a broker-less pub/sub system. This approach allows subscribers to maintain credentials according to their subscriptions. Private keys are assigned to the subscribers and are labeled along with the credentials. A publisher combines each encrypted event with a set of credentials. It uses identity based encryption (IBE) mechanisms to ensure that a particular subscriber can decrypt an event if and only if there is a match between the credentials combined with the event and the key; and to allow subscribers to verify the authenticity of received events.

II. RELATED WORK

Here, now let us considered some of the previous research work related to the traditional broker oriented pub/sub systems. Scalability and expressiveness of the system is the main focus of all these work and less attention is paid to the security. The work is as follows:

A. Sahai, J. Bethencourt and B. Waters [2], have proposed a system called as Ciphertext-Policy Attribute-Based Encryption for the complex access control strategy on encrypted data. This technique is used to keep the encrypted data secret in a situation where the storage server is not secured. In the existing Attribute-Based Encryption systems attributes are used to describe the encrypted data and also define the policies into users keys. In this proposed system users credentials are described by the attributes and a policy is determined by the party that is encrypting the data for who can decrypt the encrypted data.

E. Bertino, S. Choi, G. Ghinita [3], proposed a system in which a list of subscriptions is submitted by every user to a broker, and broker routes data from publishers to the particular subscribers. Firstly, a broker receives a notification from the publisher which has a value from a publisher; broker will forward that to the subscribers whose subscriptions have the similar value to that in the publication. In many applications, the data that is to be published is confidential, and it is necessary that its contents must not be revealed to the brokers. Also, a users subscription may also contain sensitive information that must be protected from the brokers. As a result, there arises a difficulty that how to route publisher data to the particular subscriber without disclosing the plain text values of the notifications and subscriptions to the brokers.

G. Russello, M. Ion, B. Crispo [4], proposed a publisher/subscriber system where the publisher and the subscriber are loosely coupled. Publisher creates events and sends it to the interested subscribers by using a network of brokers. Subscribers define their interest by specifying the subscription filters which is used by the brokers for the delivery of events to particular subscriber. So, it is necessary that any technique that is used for preserving the confidentiality of both the events and the subscriptions should not require the sharing of secret keys between publishers and subscribers. Also, such a technique should not prohibit the expressiveness of subscription filters and should also allow the broker to

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

handle event filtering for the routing of the events to the interested subscribers. Existing solutions available for pub/sub systems dont address these issues, so here is a mechanism that will address all these issues.

A. Iyengar, L. Liu, M. Srivatsa, [5], proposed an Event Guard which is a framework for the construction of secure wide area publisher-subscriber systems. Event Guard mechanism is not only used to provide the security guarantees but also maintains the systems over all scalability, performance and simplicity. Event Guard architecture has three main components. Security guards that are plugged-into a content-based pub-sub system is a first component, scalable key management algorithm is a second component that is useful to enforce access control on subscribers, and a publisher-subscriber network design is a third component which is capable of recovering from the difficult situations quickly.

R. Molva, A. Shikfa, M. Onen, [6], proposed a system in which privacy-preserving forwarding of the encrypted contents based on subscribers interests is handle by a set of security mechanisms. The advantages of this scheme are 1) It ensures data confidentiality of the published events 2) Privacy of the subscribers with respect to their subscriptions in a model where the subscribers, publishers and the intermediate nodes i.e. brokers in charge of data forwarding do not trust each other. This scheme uses a multi-layer encryption which allows brokers to manage forwarding tables and to perform content forwarding using encrypted content without accessing the plaintext of the data. Key sharing among the end-users is avoided by this scheme and focused on an enhanced CBPS model in which brokers can also be subscribers at the same time.

B. Maniymaran, R.S. Kazemzadeh, V. Muthusamy, A.K.Y. Cheung, G. Li, H. A. Jacobsen, [7], have done a detailed overview of a content-based publish/subscribe system called as the PADRES. PADRES is helpful in accessing data that is produced in the past and that will be produced in the future, handle network failures, correlating events, balance the traffic load among brokers. It can also filter, aggregate, correlate, and give any combination of historic and future data. They also proposed several applications that can take benefit from the content-based publish/subscribe system and also take advantage of its scalability and robustness.

A proper middleware support is needed while we are developing a large-scale distributed system that is going to be used over the Internet; which handles the communication needs of those application clients in a scalable and efficient way, and without compromising traditional middleware features. P. Pietzuch [8], have described a distributed, event based middleware called as Hermes that provides peer-to-peer communication for scalable and robust event transmission. Peer-to-peer technique is used by Hermes for managing its network of event brokers and also adding fault-tolerance to its event transmission algorithms in the publisher/subscriber systems.

S. l. Zhu, B. Yang, Y. Sun, Y. Yu [9], have proposed first identity based signcryption scheme. However, their scheme still has some security weaknesses and they further, propose a corrected version of their scheme and prove its security under the existing security model for identity-based signcryption.

M. A. Tariq, B. Koldehofe, G. G. Koch, I. Khan, and K. Rothermel [10], have proposed a peer-to-peer-based approach that satisfies the individual delay requirements of subscribers in the presence of bandwidth constraints. In this approach, subscribers are allowed to dynamically adjust the granularity of their subscriptions based on their bandwidth constraints and delay requirements. Publish/Subscribe overlay is maintained by the subscribers in a decentralized manner by establishing connections to peers that provide messages meeting exactly their subscription granularity and complying with their delay requirements.

The problem of providing privacy/confidentiality in CBPS systems is challenging task, as the solution to the above problem should permit content brokers to make routing decisions based on the content without revealing the content to them. The existing work that attempted to solve this particular problem was not successful. The problem that may appear is unsolvable since it involves conflicting goals, but in this paper, M. Nabeel, N. Shang, and E. Bertino [11], have proposed an approach to preserve the privacy of the subscriptions and confidentiality of the data that is published by the Content Publishers using cryptographic techniques when third-party Content Brokers are utilized to make routing decisions based on the content.

III. PROPOSED SYSTEM

Problem Statement

We have developed a system for providing security to the data while it is transmitting between the publisher and subscriber, it is important to manage the confidentiality and the authentication of that data in order to keep the data secured. To achieve this goal following activities will be carried out:

1. Publisher sign and encrypt events by using Identity based encryption.
2. Allow efficient routing of encrypted events.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

3. Subscribers verify the signature associated with all the attributes.
4. To ensure integrity auditing is done

The overall Block diagram of the system is shown in the Fig. 1 Function of each block is discussed below:

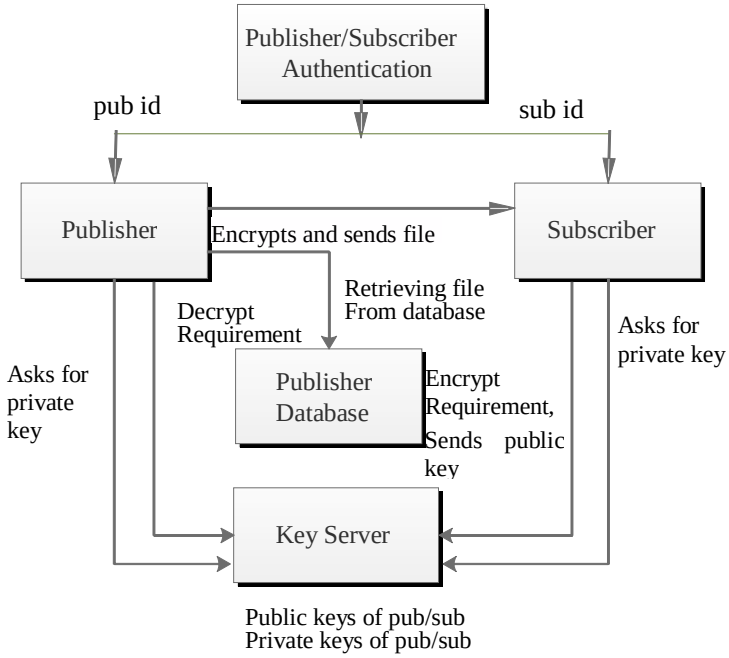

Fig. 1. Block Diagram of Proposed System

Publisher/subscriber Authentication: this method is responsible for the registration of publisher and subscriber by the key server.

Publishing Events: In this method, publisher will publish the events in the system. Publisher is authenticated by using the advertisements in which a publisher tells in advance the set of events which it intends to publish. This notification is forwarded to all the subscribers in the system and the subscribers those are interested in that particular event will send response to the publisher.

Identity based encryption: This method computes ciphertext of message by using identity-based encryption. Here, publishers and subscribers contact the key server and receive keys which are used to encrypt, decrypt, and sign the relevant messages in the pub/sub system.
Receiving event: After receiving an event, subscriber decrypts that event by using the key that is assigned by the key server.
Auditing: After receiving an event it will be check for an integrity in order to check whether given event is tampered by an attacker or not.

IV. **AIGORITHMS**

Step 1- The setup algorithm has no input other than the implicit security parameter. It gives the public parameters PK and a master key MK.

Step 2- Public parameters PK, a message M, and an access structure A are given as an input to the encryption algorithm over the set of attributes. The algorithm encrypts message M and produces a cipher text CT such that only a user who

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

has a set of attributes that satisfies the access structure will be able to decrypt the message. Here it is assumed that the cipher text implicitly contains access structure A.

Step 3- Master key MK and a set of attributes S that describe the key are given as an input to the Key Generation Algorithm. It gives a private key SK as an output.

Step 4- Public parameters PK, a private key SK, which is a private key for a set S of attributes, a ciphertext CT which contains an access policy A are given as an input to the Decryption Algorithm. If set S of attributes matches the access structure A then only the algorithm will decrypt the cipher text and return a original message M.

AES Algorithm

1. In the KeyExpansion step round keys are derived from the cipher key. AES consists of a separate 128-bit round key block for each round plus one more.
2. In Initial Round every byte of the state is combined with a block of the round key by using bitwise xor.
3. In every round following operations takes place:
    1. Substitute Bytes operation in which byte by byte substitution takes place.
    2. ShiftRows operation performs shifting of the rows of the state array where the last three rows of the state are shifted cyclically a certain number of steps.
    3. MixColumns operation performs a mixing operation which operates on the columns of the state. In MixColumns operation, combining of the four bytes in each column is done.
    4. AddRoundKey
4. Final Round has following operations:
    1. Substitute Bytes
    2. Shift Rows
    3. Add RoundKey.

V. RESULTS

For understanding and evaluation of performance of any system experimental analysis is important which give clear idea of performance of that system. For measuring the performance , we have consider parameter such as Connection delay, Encryption time, Decryption time, Attack detection time of the system. The performance of system varies for each parameter.

A.CONNECTION DELAY

Here we calculate the time required for data to travel over a network, from its source to its ultimate destination and this is called as connection delay. The Performance of the system is evaluated by calculating the average delay occurred to connect the nodes and to send message. The table 1 shows the connection delay required for message sending from one node i.e producer to other node i.e consumer. The readings are taken for the file size of range 10 kb to 50 kb. The time varies depending on file size example for 10kb the delay is 13728 ms and for 20kb require 15799 ms. After this for 30kb there is decrease in time and again for 40kb, this is due to the internet speed and traffic over network. Then till 50 kb the time varies slightly. The graph in fig. 2 shows the connection delay of producer consumer system. Here, X Axis represents the Message Size in bytes and Y Axis represents Time for connection between nodes to send message in msec. It shows that there is very little increase in time from 10kb to 20kb. Again little decrement from 20 kb to 30kb. 40kb requires 16121 msec where as 50kb requires 13059 msec. The connection delay and message sending time increases with the message size.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

Sr. No.	Message Size (bytes)	Connection delay time(msec)
1	10000	13728
2	15000	13797
3	20000	15799
4	25000	15669
5	30000	13121
6	35000	15730
7	40000	16121
8	45000	15808
9	50000	13059

Table 1: Connection Delay

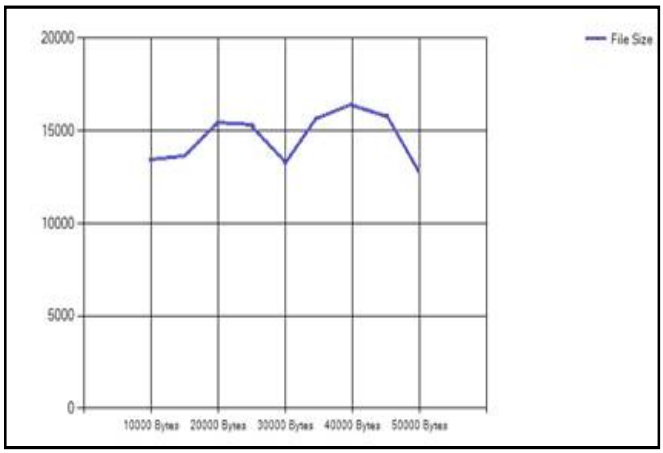

Fig. 2 Connection delay of Producer Consumer System

B. ENCRYPTION TIME

Before publishing, the message should be encrypted by the producer to ensure the confidentiality of the system. The original message is converted to the encrypted message i.e cipher text in Encryption process. The total time required to convert the plain text message into cipher text is known as Encryption time. Table 2 shows the readings obtained from experimentation. Figure 3 shows the encryption time of the system. Here, X Axis represents Message Size in bytes and y axis represents the time required for encryption. It shows that the time varies as message size changes. The time require for 10000 bytes is little less than 20000. The curve shows that from 30000 bytes to 40000 there is sudden decrement in the time. The variations are due to system processing speed and CPU utilization.

Sr. No.	Message Size(bytes)	Encryption time(msec)
1	10000	5204
2	15000	4879
3	20000	5796
4	25000	5940
5	30000	5194
6	35000	5428
7	40000	5558
8	45000	5322
9	50000	5625

Table 2: Time for Encryption

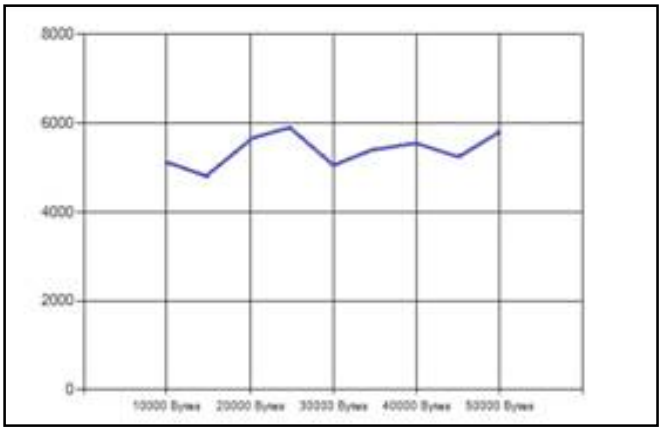

Fig. 3 Encryption Time

C. DECRYPTION TIME

The Encrypted message is converted to the original message in Decryption process. The total time required to convert the encrypted message into plain text is known as Decryption time. Table 3 shows the time taken for decryption in msec and the message size in bytes. The time required to decrypt 10000 bytes of message is 9882 ms and for 20000 bytes is 8792 ms. There is decrease in two values. The further reading shows that as the size increases the time varies slightly. The decryption time varies because of Internet speed and CPU utilization of a system. Figure 4 shows the graph of decryption time of the system. Here, X Axis represents Message Size in bytes and y axis represents the time required for decryption. It is measured in the millisecond. The curve shows than the per-formance varies by the message size. From 10000 byte to 20 bytes there is little decrement. Also from 30000 to 400000 there is increment. The variations are due to system speed , CPU utilisation.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

Sr. No.	Message Size(bytes)	Decryption time(msec)
1	10000	9882
2	15000	9734
3	20000	8792
4	25000	10812
5	30000	8912
6	35000	8673
7	40000	9103
8	45000	9846
9	50000	9243

Table 3: Time for Decryption

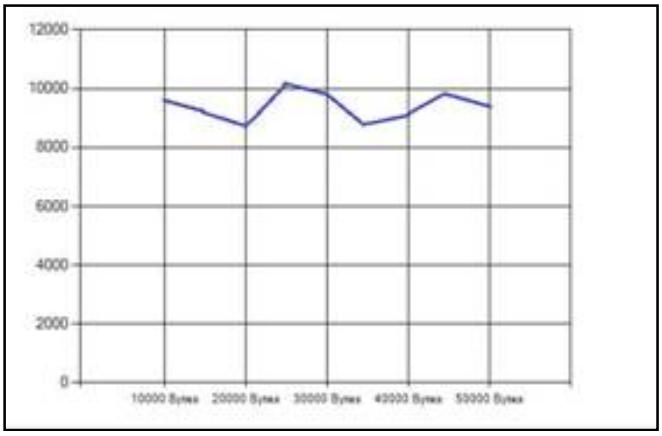

Fig 4. Decryption Time

D. RESULTS FOR AUDITING

Security monitoring on the network is important, because computers sharing data are most readily available to an attacker. Without mechanisms in place to detect attacks as they occur, an system may not realize its security. Therefore it is vitally important that computers residing in the network are carefully monitored for a wide range of audit events. The attacker tries to attempt attack on any node in a system. Once the attacker comes to know the ip address of consumer node, he will attack on that node and replaces the recent original message by his own message. Table 4 shows that attacker made attack on different nodes in a system and has replaces the original message by some other massage. The different readings has note down. It shows the number of characters are replaced by other characters. For, example here 5kb original message was having 4673 characters and the attacher directly removed this message and replaced by 1583 characters. Likewise the same case happened for 10 kb , 20 kb and 30 kb messages of different consumers. Figure 5 shows the number of characters replaced by attacker. Here, Y Axis represents Number of character in original message and characters of attackers message and X Axis represents the different consumers.

Sr. No.	Nodes	Original Message Size	Number of characters	Characters Replaced
1	consumer 1	5 kb	4673	1583
2	consumer 2	10 kb	14368	8972
3	consumer 3	20 kb	18880	1249
4	consumer 4	30 kb	26432	9342

Table 4: Number of characters replaced by attacker

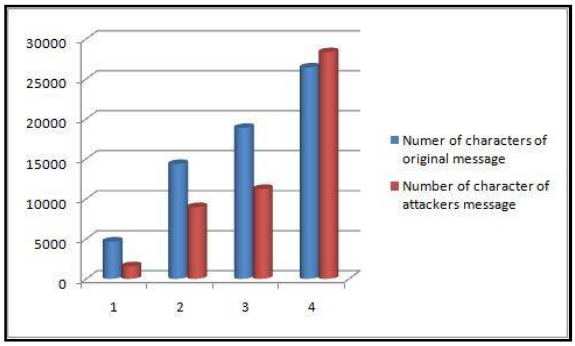

Fig. 5 Characters replaced by attacker

Detection of Attack : The system detects the attack attempted on nodes in a network. The detection of attack is done by auditing. The system comes to know about the attack. After attack takes place the attack is detected by the system and its time is calculated. Table 5 shows the attack detection time with different message size. Time required for attack detection depends on the size of message to replace. As the message size i.e number of characters increases the time required to detect also increases. Figure 6 shows the Attack Detection time. Here, X Axis represents the Attack message size in number of characters and Y Axis represents time for Attack detection(msec). As the size of message increases the attack detection time also increases.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

Sr. No.	Message Size of Attacker	Time to detect attack
1	1583 characters	16 ms
2	8972 characters	46 ms
3	1249 characters	58 ms
4	9342 characters	63 ms

Table 5 Attack Detection Time

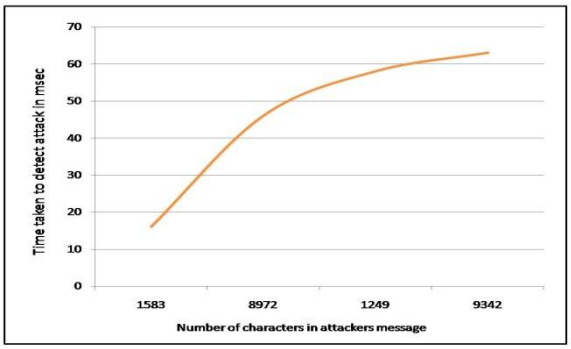

Fig 6: Attack Detection time graph

Table 6 shows the Attack detection performance. The attacker tries to attempt attack number of times and those attacks are detected by the system. If attacker attacks 5 times and system detects that 5 times then the performance is 100%. Similarly if attack is made 20 times and if 12 times it is detected then the performances of detection is 60% . In this way the performance of attack detection varies. Figure 7 shows the attack detection performance of the system. It shows that in some case 100 % result is achieved where as in some cases the result varies like 60%, 88.88%.

Sr. No.	Number of times attack attempted	Number of times attack detected	Attack detection Performance in (%)
1	5	5	100%
2	20	12	60%
3	35	26	74.28%
4	45	40	88.88%

Table 6 Attack detection performance

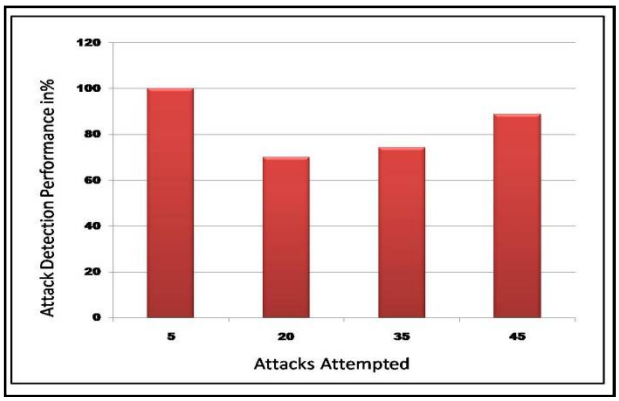

Fig. 7 Attack Detection performance

After detection of attack, the system sends notification to nodes about an attack. The system notifies to consumer that there is an attack on the system which makes the consumer alert. For preserving the data integrity, it is very important to prevent the system from attack. After detecting the attack the system saves the ip address of the attacker and it will prevent the further attack by not allowing the node to receive message again from attacker ip address. Hence attacker will not again able to attack on that nodes again once attempted. Hence, this is preventing our system from attack and improves security and data integrity.

VI. CONCLUSION

In this paper an efficient approach to provide authentication, confidentiality and integrity in a broker-less content-based pub/sub system is defined. This approach is highly scalable in terms of number of subscribers and publishers in the system and the number of keys maintained by them. Here a mechanism is developed to pass the messages between publishers and subscribers according to their subscriptions and advertisements. Private keys assigned to publishers and subscribers, and the cipher texts are labelled with credentials. Identity-based encryption is used to ensure that a particular subscriber can decrypt an event only if there is a match between the credentials associated with the event and its private keys and to allow subscribers to verify the authenticity of received events. Through all these concepts a scalable system can be created. Though there are lots of advantages in the proposed system still there is a scope to provide more efficient event routing mechanism as currently there is lots of research is going on in this area.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

REFERENCES

1. Muhammad Adnan Tariq, Boris Koldehofe, and Kurt Rothermel, "Securing Broker-Less Publish/Subscribe Systems Using Identity-Based Encryption", IEEE Transactions on parallel and distributed systems, vol. 25, no. 2, February 2014.
2. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption", Proc. IEEE Symp. Security and Privacy, 2007.
3. S. Choi, G. Ghinita, and E. Bertino, "A Privacy-Enhancing Content-Based Publish/Subscribe System Using Scalar Product Preserving Transformations", Proc. 21st Int'l Conf. Database and Expert Systems Applications: Part I, 2010.
4. M. Ion, G. Russello, and B. Crispo, "Supporting Publication and Subscription Confidentiality in Pub/Sub Networks", Proc. Sixth Int'l ICST Conf. Security and Privacy in Comm. Networks (SecureComm), 2010.
5. M. Srivatsa, L. Liu, and A. Iyengar, "EventGuard: A System Architecture for Securing Publish-Subscribe Networks", ACM Trans. Computer Systems, vol. 29, article 10, 2011.
6. A. Shikfa, M. O¨ nen, and R. Molva, "Privacy-Preserving Content-Based Publish/Subscribe Networks", Proc. Emerging Challenges for Security, Privacy and Trust, 2009.
7. H. A. Jacobsen, A.K.Y. Cheung, G . Li, B. Maniymaran, V. Muthusamy, and R.S. Kazemzadeh, "The PADRES Publish/Subscribe System", Principles and Applications of Distributed Event-Based Systems. IGI Global, 2010.
8. P. Pietzuch, "Hermes: A Scalable Event-Based Middleware", PhD dissertation, Univ. of Cambridge, Feb. 2004.
9. Y. Yu, B. Yang, Y. Sun, and S. l. Zhu, "Identity Based Signcryption Scheme without Random Oracles", Computer Standards & Inter-faces, vol. 31, pp. 56-62, 2009.
10. M. A. Tariq, B. Koldehofe, G.G. Koch, I. Khan, and K. Rothermel, "Meeting Subscriber-Defined QoS Constraints in Publish/Subscribe Systems", Concurrency and Computation: Practice and Experience, vol. 23, pp. 2140-2153, 2011.
11. M. Nabeel, N. Shang, and E. Bertino, "Efficient Privacy Preserving Content Based Publish Subscribe Systems", Proc. 17th ACM Symp. Access Control Models and Technologies, 2012.