INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

EFFICIENT AND SECURED CONTENT BASED PUBLISH/SUBSCRIBE SYSTEM TECHNIQUE

MR. AJIT. B. LINGHATE, DR. S. S. SHEREKAR, DR. V. M. THAKARE

SGBAU, Amaravati, India.

Accepted Date: 15/03/2016; Published Date: 01/05/2016

Abstract*:* In a Publish Subscribe system, distributed entities, called participants, communicate with each other by exchanging messages, often referred to as events. Participants can publish events on the system, or they can subscribe to events of their interest by specifying the type or the content of events they are interested in. A Publish Subscribe middleware routes events to subscribers, ensuring that they receive only information matching their interests. The most significant and recognized advantage of this mode of interaction is the decoupling of communicating parties in space, time, and synchronization , publishers and subscribers do not have to know each other, or in many cases do not even have to be connected to the system at the same time. It is difficult to achieve authentication of publisher and subscriber as there is loose coupling of publisher and subscriber. By using pairing based cryptography mechanism the system ensured the authentication of publishers and subscribers as well as confidentiality to the needs of a publish/subscribe system.

Keywords: Publish-Subscribe, Content-Based, Secured PUB-SUB System.

Corresponding Author: MR. AJIT. B. LINGHATE

Access Online On:

www.ijpret.com

How to Cite This Article:

Ajit B. Linghate, IJPRET, 2016; Volume 4(9): 1585-1596

PAPER-QR CODE

INTRODUCTION

The publish/subscribe(pub/sub) communication paradigm is one of the most used paradigm because of it uses decoupling of publishers from subscribers in terms of space, time, and synchronization between publisher and subscriber. Publisher published the data in the system and as per subscription the subscriber received the information. The information published by the publisher are routed to particular subscriber. Content-based pub/sub is the alternative form that provides the most useful subscription model, where subscription define restrictions on the message information. In distribution of news, stock market, traffic control, and public sensing the expressiveness of publish subscribe system is used. The pub/sub needs to provide a mechanism which accomplished the basic security demands of the applications such as access control and confidentiality. Moreover, the content of events should not be exposed to the routing information and a subscriber should receive all relevant events without knowing its subscription to the system. To solve these security issues in a content-based publish/subscribe system imposes new challenges. For instance, end-to-end authentication using a public key infrastructure (PKI) conflicts with the loose coupling between publishers and subscribers, a key requirement for building scalable pub/sub systems. For PKI, to encrypt event the publisher have to maintain the keys for particular subscriber. Subscriber should have details of the public keys of all relevant publishers to verify the authenticity of the received information.

This paper discusses on 5 methods i.e. broker-less publish/subscribe systems using identity-based encryption, quality of service in wide scale publish–subscribe-systems, infrastructure-free content-based publish/subscribe, efficient filter privacy-aware content-based pub/sub systems, PUBSUB: An Efficient Publish/Subscribe System.

II) BACKGROUND

Muhammad Adnan Tariq et al.(2014) [1]has proposed that in broker less publish subscribe system to provide authentication and confidentiality key encryption method is used. The broker less key management allows subscribers to store credentials according to their subscriptions. Private keys are assigned to the subscribers which are are named with the credentials. For each set of credentials a publisher associated with each encrypted event. Broker less publish subscribe system adapted identity-based encryption (IBE) mechanisms 1) to ensure that a particular subscriber can decrypt an event only if there is a match between the credentials associated with the event and the key; and 2) to allow subscribers to verify the authenticity of received events. In addition to, 1) extensions of the cryptographic methods to provide efficient routing of

encrypted events by using the idea of searchable encryption, 2) “Multi credential routing” a new event dissemination strategy which strengthens the weak subscription confidentiality, and 3) a detailed analysis of different attacks on subscription confidentiality.

Paolo Bellavista et al.(2014)[2] has proposed a Publish/Subscribe (PUB/SUB) messaging pattern is widely considered as a fundamental way to enable scalable and flexible communication in highly distributed systems. The most significant and recognized advantage of this mode of interaction is the decoupling of communicating parties in space, time, and synchronization: publishers and subscribers do not have to know each other, or in many cases do not even have to be connected to the system at the same-time.

Vinod Muthusamy et al.(2014)[3] has proposed large-scale pub/sub systems which are popular in industry. For example, Google and Yahoo Message Broker pub/sub systems, respectively, to integrate their Web applications; Super Montage, which is used to discriminate financial orders and data in a distributed system that uses a pub/sub model; retailers such as Target exchange supply chain information using the Global Data Synchronization Network pub/sub network ; and a pub/sub system developed by IBM is used for delivering tennis match scores to the users around the globe. Common to these applications is the selective broadcasting of data to a very large number of geographically spread entities. While large-scale pub/sub applications, like those above, can be built today, they typically require a large company’s resources to set up and manage .The proposed design virtually eliminates pub/sub cost for infrastructure. The users’ resources are automatically used to accomplish infrastructure-free scalability. In addition, the design self-organizes to adapt to bottlenecks and faults, so no personnel are required to administer the network.

WeixiongRao et al.(2013)[4] has proposed that in recent years, the content-based publish/subscribe (pub/sub) has become a accepted paradigm to decouple information producers and consumers (i.e., publishers and subscribes, respectively). It offers expressive and flexible information targeting capabilities for many Internet and mobile applications. In such a system, subscribers declare their personal interests by defining subscription conditions as filters, and publishers produce publication messages. On receiving publication messages from publishers, brokers match publications with registered filters, and forward each matched publication to needed subscribers in a one-to-many manner. The content-based pub/sub offers an excellent decoupling property with the help of brokers. Unfortunately, brokers also introduce privacy concerns . When users define their personal interests as filters and register the filters to brokers, they could receive the publications containing sensitive information (e.g., corporation

or military) or political/religious affiliations. The untrusted brokers could expose the personal and sensitive interests with respect to such users. Furthermore, by deploying brokers as public third-party servers, many modern applications, like service oriented architectures (SOAs) and social computing platforms, have adopted the content-based pub/sub paradigm. Attacks against public third-party servers could easily leak subscribers' interests.

Tania Banerjee et al.(2015) [5] has proposed Pub/Sub systems are used in diverse applications with varied performance requirements. For example, in some applications events occur at a much higher rate than the posting/removal of subscriptions while in other applications the subscription rate may be much higher than the event rate and in yet other applications the two rates may be comparable. Optimal performance in each of these scenarios may result from deploying a different data structure for the subscriptions or a different tuning of the same structure. Many commercial applications of pub/sub systems have thousands of attributes and millions of subscriptions. So, scalability in terms of number of attributes and number of subscriptions is critical.

III) PREVIOUS WORK DONE

Muhammad Adnan Tariq et al.(2014)[1] has proposed that for providing security mechanisms in pub/sub, it control the principles of identity-based encryption to support many-to-many communications between subscribers and publishers. In this approach, publishers and subscribers communicate with a key server. Credentials are assigned to the key server and in turn it receives keys which fit the articulated capabilities in the identification. Subsequently, these keys are used for encryption, decryption, and sign relevant messages in the content based pub/sub system, i.e., the credential becomes authorized by the key server. A credential consists of two parts: 1) a binary string which describes the capability of a peer and 2) its identity proof. The last is used for authentication against the key server and verification whether the capabilities equivalent to the identity of the peer. While this can happen in a variety of ways, for example, relying on challenge response, hardware support, and so on. Subsequently, it uses the term credential only for referring to the capability string of a credential.

Paolo Bellavista et al.(2014)[2] has proposed a PUB/SUB middleware is a distributed platform that allows its participants to exchange information with each other in the form of events. Without loss of generality, the middleware assume an event to be a set of key-value pairs, whose meaning is generally application-dependent. A participant enters information in the system by publishing; it can also express interest in particular events, by means of subscriptions. The middleware delivers events to subscribers according to their subscriptions. In order to subscribe

to events, participants select subspaces of the space of all possible events by providing one or more subscription filters, which, in their more general form, are boolean predicates on the event fields. Whenever an event is published, the system will dispatch it to a set of subscribers that have specified a subscription filter that matches the event(i.e., it evaluates to true when applied to the event).

VinodMuthusamy et al.(2014) [3]has proposed PUB/SUB is a data dissemination model with three entities: The publisher which produced the information, the subscriber which receive the information , and the broker which is mediator between the publisher and subscriber. There may be one broker or a set of distributed brokers. For example, in a stock quote dissemination application, the publisher would be the stock exchange, and the consumer could be a stock broker interested in tracking certain stocks. A subscriber expresses his interest in these stocks by sending a subscription message to the broker. The publisher communicates the latest stock updates by sending a publication message to the broker. Upon receipt of , the broker forwards to those subscribers with matching subscriptions. In content-based pub/sub, publications consist of a set of attribute, value pairs, and subscriptions are typically conjunctions of attribute, operator, value tuples, where the operator can be one of and include operations over strings. This allows subscriptions to discriminate based on the content of the publications.

Weixiong Rao et al.(2013)[4] has proposed that Publishers (the users or associated software gents that produce publications) first announce advertisements of to-be-published messages to brokers, and then publish content messages. Subscribers (the users or associated software agents that consume publications)declares their interests by filters, and send subscription requests containing the filters to brokers. Brokers decouple publishers and subscribers to offer asynchronized content delivery. On receiving advertisements from publishers, brokers validate filters and then organize filters into a filter indexing structure, for example, a partially ordered set (in short poset). Next, when publications come, with the help of the poset, the brokers match incoming publications with the indexed filters. After matched filters are found, the brokers forward publications to the associated subscribers.

Tania Banerjee et al.(2015) [5] has proposed a PUB/SUB , which is a versatile and scalable, content-based pub/sub system that may be tuned to provide high performance for diverse application environments. PUBSUB is versatile because its architecture supports a variety of predicate types (e.g., ranges, regular expressions, string relations) as well as a heterogeneous collection of data structures for the representation of subscriptions in order to achieve high throughput. The performance of a version of PUBSUB that was tuned for applications in which

events occur far more frequently than subscription posting/deletion is compared with the performance of the pub/sub systems BE Tree .

IV) EXISTING METHODOLOGIES

In broker less publish subscribe system approach, publishers and subscribers communicate with a key server. Credentials are assigned to the key server and in turn it receives keys which fit the expressed capabilities in the credentials. Subsequently, these keys are used for encryption, decryption, and sign relevant messages in the content based pub/sub system.The cipher text, are assigned with credentials and the keys are assigned to publisher and subscriber. If there is a match between the identification of the cipher text and the key the particular message gets decrypted . For each authorized credential publisher and subscriber are assigned private key. The public keys are generated by a string concatenation of a credential, an epoch for key revocation, a symbol (SUB:PUB) which distinguished each publishers from subscribers. There is no need to contact the key server for generating the keys for the communicating system. Similarly, it does not require any middleware for encryption and decryption of event.

In QoS-based services, the involved parties usually perform a quality agreement process to determine the exact service level to be needed at runtime. This process, in the context of PUB/SUB systems, has been often modeled with a publisher offered – subscribed requested (PO-SR) pattern: publishers defines a set of quality properties which they are going to offer for their publications, while subscribers request to the publisher for the desired service level for the delivery of their events. In the view, by concentrating only on participants, this simple agreement model fails at capturing the fundamental role that the middleware has in this process. In fact, in many cases, the middleware distributed components (e.g., the overlay brokers) must have and possibly reserve a nonnegligible amount of computing resources to provision service with guaranteed quality. When a publisher performs a publish action, it can also providea QoS specification describing the offered QoS. Similarly, advertise actions allow a publisher to declare beforehand the QoS properties it intends to offer for its events, and subscribe actions let a subscriber specify its required quality level.According to this model, for events to match a subscription, it is not sufficient that they satisfy the corresponding subscription filter, but, in addition, the requested and offered quality properties must be compatible, and the middleware must confirm the QoS agreement, possibly allocating the necessary resources.

In Infrastructure-Free content-based Publish/Subscribe, it map the pub/sub matching problem to a distributed multidimensional indexing problem. In particular, publications and subscriptions are mapped to regions in a multidimensional b space such that the intersection of these region

simplifies a match of the corresponding publications and subscriptions. The multidimensional space is recursively partitioned into regions and indexed by a search tree, nodes of which are managed by peers in the network. The indexed regions, as well as subscriptions and publications, are uniquely labeled with keys,which serve to identify peers in the network that manage the corresponding search tree nodes. The keys are designed to allow a search tree node to easily determine the keys of its parent and child nodes, which, again, serve as keys to the underlying DHT to find the relevant peers for these nodes.

The anonymizer in fig 1 engine accepts advertisements from publishers (step a) and filters from subscribers (step b:1). The purpose of the anonymizer engine is to cloak incoming (real and uncloaked) filters and to output cloaked filters that are used to protect real filters (step b:2). After that, the broker maintains the indexing structure for the cloaked filters, and forwards publications to matched subscribers (steps c and d). Since the broker registers cloaked filters (instead of real filters), the matching publications contain redundant ones. Thus, among the received publications, subscribers need to filter out those publications that do not match real filters, and alert users of only the matching publications.

Figure 1: Privacy aware publish subscribe system

Fig. 2 gives the organization of the subscription database used in PUBSUB. This database comprises a collection of level-1 attribute structures A1,,,Am, where m is the number of attributes. It assume that the allowable attributes have been numbered 1 through m and that the attributes in a subscription are ordered using this numbering of attributes. The attribute structure Ai stores all subscriptions that include a predicate on attribute i but not on any attribute j < i. The attribute i is associated with the structure Ai. With the assumptions on attribute ordering within subscriptions, Ai contains all subscriptions whose first attribute is i. In practice, many of the A is will be empty and only non-empty attribute structures are stored in PUBSUB. The distribution of subscriptions across these buckets is determined by the attribute i predicates in these subscriptions and the data structure D used for keeping the track of the buckets. For uniformity, level-1 attribute structures are associated with a header bucket that is always empty.

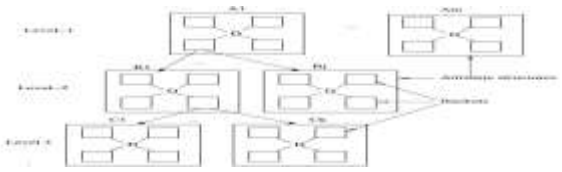


Figure 2: PUB-SUB Organization

V) ANALYSIS AND DISCUSSION

Pub/Sub is a data dissemination model with three entities: 1) Publisher 2) Broker 3) Subscriber

1) Publisher: Publisher is the data producer, publisher provides the data to the related subscribers

2) Broker: Broker mediates between the Publisher and Subscribers. There may be one broker or a set of distributed brokers.

3) Subscriber: Subscriber is the consumer, subscribers subscribes to related publisher and receives the data produced by publisher.

Publish Subscribe Systems	Advantages	Disadvantages
Broker less publish subscribe system	It minimises overload of server to keep keys.	Keys management for peer system is tedious and error prone.
Quality of Service in wide scale publish–subscribe-systems	It provides security to the content of the publisher and other services	The model is implemented in the Guaranteed Delivery protocol, which requires a specific mapping between the information flow graph and the broker topology hence requires broker server between the publisher and subscriber

Infrastructure-free content-based publish/subscribe System	Server key management task is reduced.	Publisher and subscriber have to be authenticated by themselves and hence leads to error in security.
Efficient filter privacy-aware content-based pub/sub systems	In this technique filter is used which filters out unnecessary information to the subscriber.	Subscribers have to be aware of the content of the publisher to filter unnecessary information.
PUB/SUB-An Publish/Subscribe System	PUBSUB is faster than BE-Tree by 65-80 percent	Pub/Sub systems have thousands of attributes and millions of subscriptions. So, scalability in terms of number of attributes and number of subscriptions is critical.

TABLE 1:Comparison between different Publish Subscribe System.

VI) PROPOSED METHODOLOGY

A pub/sub environment is a communication paradigm for supporting many-to many communication (refer to Fig. 3). At the core of the pub/sub system there is a set of servers. This set of core servers is called the backbone of the pub/sub system. Any sender willing to send a data contacts a server in the backbone and sends the data to this server. Any receiver willing to receive that data contacts an appropriate server in the backbone and reads the data.

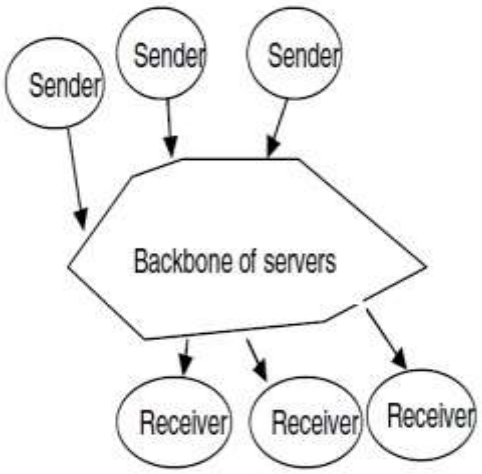
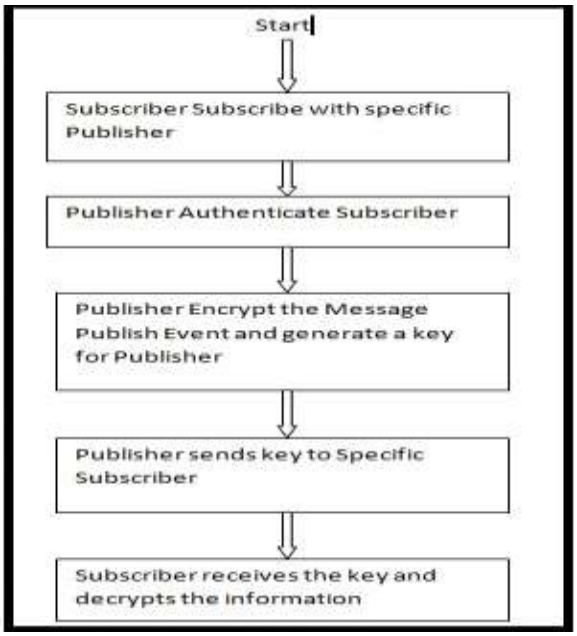


Figure 3: Publish Subscribe System



Flowchart 1: Flow Chart Of Secured publish Subscribe System

Algorithm of Secured publish subscribe system

Step1:

Subscriber Subscribe with Specific Publisher for Receiving information.

Step2:

After receiving request from Subscriber Publisher authenticate Subscriber for receiving information.

Step3:

Publisher Publish information in encrypted form and send a key to decrypt the information to subscriber.

Step4:

Subscriber receive key and decrypt the information.

VII) OUTCOME POSSIBLE RESULT

In Publish Subscribe System security of the content published by the publisher is the main issue, to handle this issue encryption of the message is performed by using secret key. As encryption is performed the content becomes secure and the proposed method tries to minimize the risk of loss of information by using a cryptographic algorithm.

VIII) CONCLUSION

This paper focused on content based secure Publish Subscribe system and analyses different methods for publishing the content which are broker-less publish/subscribe systems using identity-based encryption, quality of service in wide scale publish–subscribe-systems, infrastructure-free content-based publish/subscribe, efficient filter privacy-aware content-based pub/sub systems, PUBSUB: An Efficient Publish/Subscribe System. In this paper the proposed technique is a effective methodology for transmission of secured content by publisher to subscriber as there are subscriber authenticated by the publisher the transmission of key and information overhead gets minimized and this system can be used in weather fore-casting, in social networking, in bulletin board system, etc.

IX) FUTURE SCOPE:

From Observation, the scope and planned to be studied in future work, the proposed algorithm secures the content of publisher by encryption it. In future there can be a small key used for cryptographic function so that overhead of transferring a key gets minimised and transmission speed of content from publisher to subscriber increased.

REFERENCES

1. Muhammad Adnan Tariq and Kurt Rothermel "Securing Broker-Less Publish/SubscribeSystems Using Identity-Based Encryption" *IEEE Transactions On Parallel And Distributed Systems*, Vol. 25, No. 2, PP.518-528,FEBRUARY 2014.
2. Paolo Bellavista and Andrea Reale "Quality of Service inWide Scale Publish–Subscribe Systems" *IEEE communications surveys tutorials*, vol. 16, no. 3, PP.1591-1616 third quarter 2014.
3. VinodMuthusamy and Hans-Arno Jacobsen" Infrastructure Free Content-Based Publish/Subscribe" *IEEE Transactions On Networking*, Vol. 22, No. 5, PP.1516-1530 October 2014.
4. WeixiongRao and SasuTarkoma" Toward Efficient Filter Privacy-Aware Content-Based Pub/Sub Systems" *IEEE Transactions On Knowledge And Data Engineering*, Vol. 25, No. 11, PP.-2644-2657 November 2013.
5. Tania Banerjee and SartajSahni"PUBSUB: An Efficient Publish/Subscribe System" *IEEE Transactions On Computers*, Vol. 64, NO. 4,PP.1119-1132, April 2015.