IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727
PP 21-27
www.iosrjournals.org

Enhancing Security in Broker-Less Publish/Subscribe Systems

Reshma Rajan[1], Vidhya P M[2], Arun K Govind[3]
[1]*Department of CSE, SNGCE, Kadayiruppu, Kerala, India reshma.mariya333@gmail.com*
[2]*Department of CSE, SNGCE, Kadayiruppu, Kerala, India vidhya.mohanan@gmail.com*
[3]*Department of CSE, SNGCE, Kadayiruppu, Kerala,India arunkgovind@outlook.com*

***Abstract**: Publish/subscribe system is a wide-area communication infrastructure which allows data distribution across unlimited number of publishers and subscribers. Both publishers and subscribers have the ability to express their interest in form of events, or a pattern of events by sending the subscription to the publish/subscribe network. The provisioning of basic security mechanisms such as authentication and confidentiality is highly challenging in a content based publish/subscribe system. This issue raises a challenging requirement for handling encrypted data for the purpose of routing based on protected content. The considerable data in publish/subscribe model needs to be disseminate accurately to the interested users quickly, so improving the match efficient is a very important method to solve this problem. This paper provides authentication and confidentiality in broker-less publish/subscribe systems, by using attribute set based encryption (ASBE) and improving event matching using Predicate Priority based Event Matching, All over system is seen to be efficient and guarantees good results.*

***Keywords:** Attribute set based encryption, Broker-less, Content based publish/subscribe(CBPS), Predicate priority based event matching, Publish/Subscribe(pub/sub).*

I. Introduction

A distributed system is a software system consists of a collection of autonomous computers, connected through a network, communicate and coordinate their actions by passing messages. Publish subscribe system is one of the distributed systems, whichis an asynchronous communication model where senders, known as publishers, and receivers, known as subscribers. The messages are exchanged in a loosely coupled manner, without establishing a direct contact between them. The messages generated by the publishers are called events. Many-to-many communication paradigm and loose coupling are the major strengths of publish/subscribe system. So publishers need not know the recipients of their data and subscribers need not know the number and location of publishers. Content-based pub/sub is a variant that provides the most expressive subscription model, where subscriptions define restrictions on the message content. The asynchronous nature and expressiveness are mainly useful for large scale distributed applications. Traditionally they were using broker networks for routing of events from publishers to subscribers. In more recent systems, broker-less routing infrastructure is used. Publish/subscribe system needs to provide supportive mechanisms for basic security demands of these applications such as access control and confidentiality

Access control in pub/sub system means only authenticated publishers are allowed to distribute events and only authorized subscribers are allowed to receive that events .Contents of events are kept as confidential and events are received by subscribers without informing their subscriptions for the system. Both publication and subscription confidentiality is required to reduce risk of leakage of events in systems. For that purpose a secret key need to be shared between publishers and subscribers. public key infrastructure can be used but , which is not desirable because it would weaken the decoupling property of the model. Traditional methods of encrypting all message violates the approach of content based system. Hence a new method is needed to route events to subscribers without knowing their subscriptions and authenticating them.

In the past, most of the researches have mainly focused only on providing expressive and scalable publish subscribe systems, but little attention has been given for the need of security. Existing approaches mostly rely on the presence of a traditional broker network. These either address security under restricted expressiveness or rely on a network of (semi-)trusted brokers. Furthermore, existing approaches use coarse-grain key management and cannot provide fine-grain access control in a scalable manner. This paper propose to design a system for enhancing security in broker-less publish subscribe system by using a three-level hierarchical key structure with Ciphertext-Policy Attribute-Set-Based Encryption (ASBE) and improving the event matching using a Predicate Priority Based Event Matching algorithm.

II. Literature review

A publish subscribe system allows information distribution from event producers i.e. publishers to event consumers i.e. subscribers. These publish subscribe systems having different types of infrastructure including topic based systems and content based systems[1]. The flexibility of publish-subscribe comes on the other hand with a high cost in increased exposure in terms of data security and privacy. Apart from classical data security concerns such as the confidentiality and integrity of messages, the authentication of the source, access control and authorization of subscribers, publish-subscribe also raises new challenges inherent to the collapsed forwarding scheme that is the underpinning of publish-subscribe [2].

H.-A . jacobsen, A.K.Y. Cheung, G . Li, B. Ma niymaran [3] describes content based systems: Content-Based Publish/Subscribe (CBPS) consists of two end users: one is Publishers which publish information in the form of event notifications and the second is Subscribers which express their interests in certain content in the form of subscription filters. The CBPS infrastructure composed of brokers (intermediate nodes) whose task is to disseminate notifications sent by publishers to the interested subscribers.

Ying Liu, Beth Plale [4] discuss about topic based systems: In topic based systems, communication infrastructure maintains a logical channel also called as topics. A publisher publishes messages to topic. The subscriber subscribes to topics of their interests. They receive messages coming from their subscribed topic. Different subscribers subscribing to same topic will receive same messages. Subscription targets a group, channel, or topic, and the user receives all events that are associated with that group. Brokering a connection between publishers and subscribers is the act of connecting a channel supplier with a channel consumer. The enhancement in the logical channel changed the way to implement public subscribe systems.

D. Boneh, G.D Crescendo, R.Ostrovsky, and G.Persiano [5], study the matter of looking on information that's encrypted employing a public key system. Consider user Bob United Nations agency sends email to user Alice encrypted below Alice's public key. An email gateway desires to check whether or not the e-mail contains the keyword "urgent" in order that it may route the email consequently. Alice, on the opposite hand doesn't want to provide the entryway the flexibility to decrypt all her message. This paper tend to check with this mechanism as Public Key secret writing with keyword Search. As another example, contemplate a mail server that stores various messages publically encrypted for Alice by others. Victimization our mechanism Alice will send the mail server a key that may alter the server to spot all messages containing some specific keyword, however learn nothing else. This tends to outline the construct of public key secret writing with keyword search and provide many constructions.

L. Opyrchal et al. [6] proposed Group Key Approach in which subscribers have proper group keys by using several cashing approaches. Several events should be delivered to the same subset of subscribers. Such subsets are the groups with same key. Symmetric cryptographic scheme is used here and it is only applied to notifications and there is no matching on encrypted notifications. There is no scalable key management. Access control and digital signature is not used.

L. Fiege et al.[7] proposed an approach-Tunneling method that keeps the communication within the group separate from outsiders, and use credentials to establish mutual trust among the group members. The group consists of all the subscribers interested to a given topic, and the publishers relative to this topic. Client access control is provided by Attribute Certificates, hold client identity and a set of attributes. The certificate specifies that client is authorized to subscribe or publish certain events. Such a certificate can be issued by the administrator or by another trusted attribute authority. Tunneling method is used to transport notifications through untrusted brokers by encrypting the notification content. Digital signature is not used here.No scalable key management.

A. Shikfa, M. O¨nen, and R. Molva [8], in their work, they have described a set of security mechanisms that will allow for privacy-preserving forwarding of the encrypted contents based on subscribers interests. The main advantages of this scheme is, it ensures both data confidentiality with respect to the publishers and the privacy of the subscribers with respect to their interests in a model where the publishers, the subscribers and the intermediate nodes i.e. brokers in charge of data forwarding do not trust each other. The scheme depends on a multi-layer encryption that allows intermediate nodes to manage forwarding tables and to perform content forwarding using encrypted content and based on encrypted subscriber messages without accessing the plaintext of the data. This scheme also eliminates key sharing among the end-users and targets an enhanced CBPS model where brokers can also be subscribers at the same time.

The notion of attribute based encryption was first introduced by Sahai and Waters [9] as a new method for fuzzy identity-based encryption. The main goal for these models is to provide security and access control. It provides flexibility, scalability and fine grained access control. In classical model, and this can be achieved only when user and server are in a trusted domain. But what if their domains are not trusted or not same? So, the new access control scheme that is Attribute Based Encryption (ABE)' scheme ABE schemes are classified into key-policy attribute- based encryption (KP-ABE) and ciphertext-policy attribute- based encryption (CP-ABE), depending how attributes and policy are associated with ciphertexts and users' decryption keys. Several efforts followed in the literature to try to solve the expressibility problem. In the ABE scheme, ciphertexts are not encrypted to one particular user as in traditional public key cryptography. Rather, both ciphertexts and users' decryption keys are associated with a set of attributes or a policy over attributes. A user is able to decrypt a ciphertext only if there is a match between his decryption key and the ciphertext.

V. Goyal, O. Pandey, A. Sahai, and B.Waters, said that In a KP-ABE scheme [10], a ciphertext is associated with a set of attributes and a user's decryption key is associated with the tree access structure. Only if the attributes associated with the ciphertext satisfy the tree access structure, can the user decrypt the ciphertext.

J. Bethencourt, A. Sahai, and B. Waters [11] introduced CP-ABE (cipher text-policy attribute-based encryption) to encrypt the data which can be kept confidential even if the storage server is untrusted. In a CP-ABE scheme, the roles of ciphertexts and decryption keys are switched; the ciphertext is encrypted with a tree access policy chosen by an encryptor, while the corresponding decryption key is created with respect to a set of attributes. As long as the set of attributes associated with a decryption key satisfies the tree access policy associated with a given ciphertext, the key can be used to decrypt the ciphertext. However, basic CP-ABE schemes are far from enough to support access control in modern enterprise environments, which require considerable flexibility and efficiency in specifying policies and managing user attributes. In a CP-ABE scheme, decryption keys only support user attributes that are organized logically as a single set, so users can only use all possible combinations of attributes in a single set issued in their keys to satisfy policies. To solve this problem introduced ciphertext-policy attribute-set-based encryption

Hao Wang, Zhihua Zheng, Lei Wu[12] introduced hierarchical identity based encryption(HIBE), is the hierarchical form of a single identity based encryption. In an identity-based encryption (IBE) scheme, an arbitrary key is used as the key for data encryption and for decryption, a key is mapped by a key authority. In a regular IBE scheme; there is only one private key generator that distributes private keys to each users, having public keys are their primitive ID (PID) arbitrary strings. There also includes a trusted third party or root certificate authority that allows a hierarchy of certificate authorities: Root certificate authority issues certificates for other authorities or users in their respective domains. The original system does not allow for such structure. However, a hierarchy of PKGs is reduces the workload on root server and allows key assignment at several levels.

R. Bobba, H. Khurana, and M. Prabhakaran, describe[13], several possible solutions with plain CP-ABE, but none of them is satisfactory. However, using ASBE , we can solve the problem simply by assigning multiple values to the group of attributes in different sets. For each course the student has taken, he gets a separate set of values for the attributes . In this way, ASBE can enforce efficient ciphertext policy encryption for situations where existing ABE schemes are inefficient. Furthermore, ASBE's capability of assigning multiple values to the same attribute enables it to solve the user revocation problem efficiently, which is difficult in CP-ABE. ASBE is an extended form of CP-ABE which organizes user attributes into a recursive set structure. ASBE can enforce dynamic constraints on combining attributes to satisfy a policy, which provides great flexibility in access control. In the recursive attribute set assigned to a user, attributes from the same set can be combined freely, while attributes from different sets can only be combined with the help of translating items.

K. J. Gough and G. Smith [14], introduced matching algorithms based on the tree structure which translate the subscriptions into trees. These algorithms are fast because they store the predicates in such a redundant way so that they could efficiently figure out the matched subscriptions, but their space complexity is too large to be suitable for large-scale systems.

J. M. Chen, S. G. Ju, J. G. Pan, Z. W. Zu, Z. Y. Gong [15], introduced fast event matching algorithm based on content which is based on table structure. This paper presented an efficient and applied matching algorithm that uses multi-dimensional indexing mechanism to speed up constraints query and exploits the covering relation between constraints to reduce unnecessary matching. Y. Zhao, and J. Wu [16], introduces an approach towards approximate event processing in a large-scale content-based Network, which is a table match content-based event matching and forwarding engine. This employs approximate matching to provide fast event

matching against an enormous amount of subscriptions. To this end, the paper uses a hierarchical indexing table to store subscriptions. In both the papers the cost of maintaining the subscriptions in the table structures is very low in these algorithms. However, these algorithms only use the predicate's cover relationship to improve the efficiency but ignore other properties, such as the matching order of predicates. X. Guo, J. Wei, D.[17] al have proved that different matching order of predicates in one subscription would affect the efficient of the event matching process .

III. Analysis of Problem

Publish subscribe systems are provided by most researchers but less consideration is given on security of publish subscribe systems. In the past, most research has focused only on providing expressive and scalable pub/sub systems, but little attention has been paid for the need of security. Existing approaches toward secure pub/sub systems mostly rely on the presence of a traditional broker network [11]. These either address security under restricted expressiveness, or rely on a network of (semi-)trusted brokers An important privacy requirement in content-based publish-subscribe is the confidentiality of the messages through which subscribers inform the network about their interests. While encryption of these messages appears to be a suitable solution for subscriber privacy, such encryption operation raises an additional challenge for the forwarding mechanism. So the event forwarding and event matching need to be improved.

Specifically the problem statement is to secure the publish subscribe system through more improved encryption technique and improving the event matching efficiency.

3.1 Disadvantages Of Existing System

The major limitation of the existing method, using Identity Based Encryption[18] is, it can use any arbitrary string as the public key and is generated based on a single identity of user, which is not much secure. So go for Attribute Set Based encryption in which key generation is not only based on single identity of user but also based on a set of attributes. In CP-ABE, private keys can only support user attributes that are organized logically as a single set. So CP-ABE is not enough to support access control in modern enterprise environments, which require considerable flexibility and efficiency in specifying policies and managing user attributes. There is no method for improving the speed and efficiency of event matching in the existing system.

IV. Predicate priority based event matching

A subscription is composed of a set of predicates {p1, p2,_, pn}. Predicates and their Ids are stored in predicate priority table. Repetitive predicates are stored only once, because multiple subscriptions may include the same predicate. Each row in the table is organized according to predicate's data type, attribute name and comparison operator. From up to down priority (row1) ≥ priority (row2) ≥…≥ priority (row n).Predicates of the same subscription is organized into a linked list called predicate linked list. The linked list only stores predicate Id. The subscription IDs are stored in the list in sequence called subscription list. The program generates a unique ID for each subscription automatically, which is used to identify the subscription. To maintain the subordination relation of the predicates and subscriptions, every subscription has a pointer which points to the corresponding address of the predicate linked list. There are two stages of procedures, preprocessing procedure and matching procedure.

Preprocessing procedure*:* In order to organize the subscriptions into the data structure of the algorithm, we need a pre-processing before matching.

```
Input: All Subscriptions SubList
1 For each subscription S € SubList do
2 // Step 1: Assign an ID for S
3          SubID_list=ID(S)
4          preList.clear()
5          For each predicate p €  predicates[S] do
6                    p.Id=ID(p)
7 // Step 2: Insert(predicate, Id) of S into predicate          table
 8 // push the predicate's Id into the  predicate linked list
 9          InsertPredicate(p,predicate_priTable)
10          InsertPreId(preList, Id)
11     End for
12 // Step 3: Add the address of predicate linked
```

```
13 // list to the corresponding subscription
14        SubID_list[S].add_address(preList)
15End for
```

Matching procedure: Next stage after preprocessing is the matching procedure. Set the pi as all the predicates set in row i, p_{i_match} represents the predicate that matches the new event, matchSubs stores the matched subscription IDs. When a new event comes, the matching process is as follows:

Input: A new Event E and predicate priority table Table[]
Output:matchSubs

```
 1 matchSubs:={}
 2 For each predicate pe in E do
 3 // Step 1 Initial predicate linked list
 4         curPrelist=preList
 5         i=findRow(Table[], pe )
 6         pi _Table[i]
          // pi represents the predicate in row i
 7 // Step 2 Find match predicates in current row
 8       p_i _match =FindMatchedPre( pi , pe )
 9              For each predicate pi in p_i _match do      10 // Step 3 Delete the pi in corresponding  curPrelist
11                      if pi ∈S.curPrelist then
12                              S.curPrelist.pop( pi )
13 // Step 4 Judge the subscription S whether              14 //match or not.
15                      if S.curPrelist.empty()then
16                              matchSubs.add(S.ID)
17              End for
18 End for
```

V. Attribute Set Based Encryption

Ciphertext Policy Attribute Set Based Encryption (CP-ASBE) - a new form of CP-ABE - which, unlike existing CP-ABE schemes that represent user attributes as a monolithic set in keys, organizes user attributes into a recursive set based structure and allows users to impose dynamic constraints on how those attributes may be combined to satisfy a policy.

5.1 Key structure

A recursive set based key structure is used, where each element of the set is either a set itself (i.e. key structure) or an element corresponding to an attribute. We define a notion of depth for this key structure, which is similar to the notion of depth for a tree that limits this recursion. The key structure can be represented as $A=\{A_0,A_1\ldots A_m\}$,Ai represents the i^{th} subset in A. The key structure defines unique labels for its subsets. Thus if there are m subsets at depth 2 then a unique index i where $1 \leq i \leq m$ is assigned to each subset. The depth of key structures that can be supported by the scheme is a system parameter that should be decided at the time of setup. Individual attributes inherit the label of the set they are contained in and are uniquely defined by the combination of their name and their inherited label.

5.2 Access Tree Policy

The ASBE scheme consists of 4 basic algorithms:

• Setup (d=2): Here d is the depth of key structure. Taking as input a depth parameter d, the algorithm outputs a public key PK and master secret key MK.

• KeyGen (MK, A, u) : $A=\{A_0,A_1\ldots A_m\}$ is a key structure, and u is the identity of a user. Taking as input the master secret key MK, the identity of user u , and a key structure A , the algorithm outputs a secret key DKu for user u .

• Encrypt(PK, M, T): M is the message, T is an access tree. Taking as input the public key PK , M and T , the algorithm outputs a ciphertext CT .

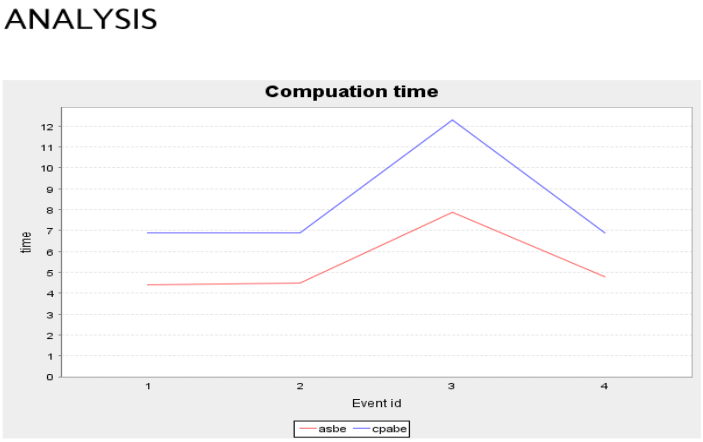
• Decrypt(CT,DKu) : Taking as input a ciphertext CT and a secret key DKu for user , the algorithm outputs a message M* . If the key structure A that associated with the secret key DKu satisfies the access tree T that associated with the ciphertext CT , then M* is the original correct message M . Otherwise, M* is null.

5.3 System Model

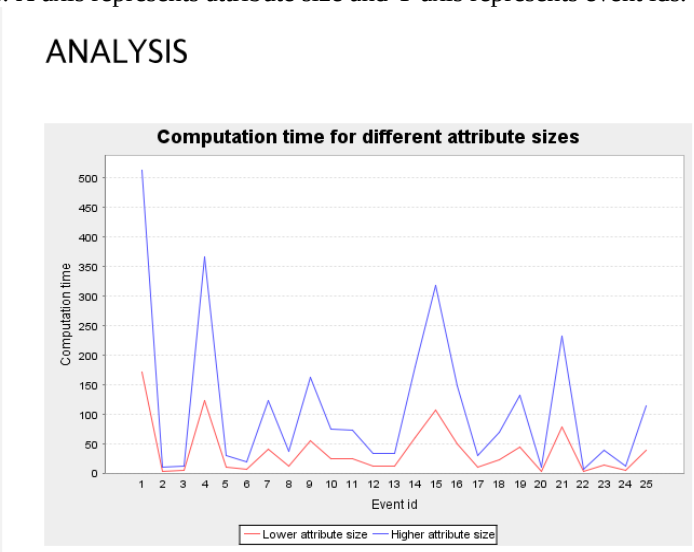
In the system a CW(ControlWord) assigned for every authorized publishers and it is updated frequently for security. CW is encrypted by AK (authorization key) and AK is in turn encrypted with an access tree policy by ASBE. Each authorized subscriber is assigned a DK and as long as the subscriber's DK can satisfy the access tree policy associated with the encrypted AK, the subscriber can decrypt the AK. Then he can decrypt CW with AK. AK and DK are managed in a different way using ciphertext-policy attribute-set-based encryption (ASBE). AK is encrypted with an access tree policy but not to a particular subscriber. Each subscriber gets a DK, which works as the secret key in ASBE. The subscriber's DK depends on his subscriptions. For each of the subscriptions, an appropriate subset is added to the subscriber's key structure. Then generate a DK according to the key structure and assign it to the subscriber.

VII. Analysis

When analysis the performance, it is seen that the system provides good performance.
The graph below shows the computation time for CPABE and ASBE schemes. The X axis represents the events published with unique ids and Y axis represents time.



When analyzing predicate priority based event matching, the graph below shows the computation time for different attribute sizes. X axis represents attribute size and Y axis represents event ids.



VI. Conclusion

This paper describes methods for enhancing authentication and confidentiality in broker-less pub/sub system. Cipher text policy attribute set based encryption is used for achieving security by encrypting the events in publish subscribe system. Since the keys used for encryptions are not only based on the identity of a user but also based on a set of attributes, it is more secure than identity based encryption. Moreover a predicate priority based event matching algorithm is used to improving the event match efficient. Changing the order of predicates based on their priority in content-based pub/sub system got better performance.

References

[1] Minakshi B. Shingan, Sanchika A. Bajpai, "Review: Securing Broker-Less Public/Subscribe Systems Using Identity-Based Encryption", International Journal of Science and Research (IJSR), Volume 3, Issue 11, November 2014.
[2] 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.
[3] H.-A . jacobsen, A.K.Y. Cheung, G . Li, B. Ma niymaran, V . Muthusa my, and R.S. Ka zemzadeh, "The PADRES Publish/Subscribe System," Principl es and Applications of Distributed Event-Based Systems. IGI Global, 2010
[4] Ying Liu Beth Plale, "Survey of Publish Subscribe Event Systems", Computer Science Dept. Indiana University Bloomington, IN 47405-7104
[5] D Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," Proc. Int'l Conf. Theory and Applications of Cryptographic Techniques on Advances in Cryptology (EUROCRYPT), 2004.
[6] L.Opyrchal and A. Prakash, "Secure distribution of events in contentbased publishsubscribe systems," Proceedings of the 10th USENIX Security Symposium, August 2001
[7] L. Fiege, A. Zeidler, A. Buchmann, R. K.-Kehr, and G. M¨uhl, "Security aspects in publish/subscribe systems," Proceedings of the Third International Workshop on Distributed Event-Based Systems, 2004.
[8] A. Shikfa, M. O¨ nen, and R. Molva, "Privacy-Preserving Content- Based Publish/Subscribe Networks," Proc. Emerging Challenges for Security, Privacy and Trust, 2009.
[9] A. Sahai and B. Waters, "Fuzzy identity based encryption," in Proc. Acvances in Cryptology—Eurocrypt, 2005, vol. 3494, LNCS, pp.457–473.
[10] V.Goyal, O. Pandey, A. Sahai, and B.Waters, "Attibute-based encryption for fine-grained access control of encrypted data," in Proc. ACM Conf. Computer and Communications Security (ACM CCS), Alexandria, VA, 2006.
[11] J.Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in Proc. IEEE Symp. Security and Privacy, Oakland, CA, 2007
[12] Hao Wang, Zhihua Zheng, Lei Wu "Hierarchial Identity-Based Encryption Scheme from Multilinear Maps" 2014 Tenth International Conference on Computational Intelligence and Security
[13] R.Bobba, H. Khurana, and M. Prabhakaran, "Attribute-sets: A practically motivated enhancement to attribute-based encryption," in Proc ESORICS, Saint Malo, France, 2014.
[14] K. J. Gough and G. Smith, "Efficient recognition of events in distributed systems," In Proce edings of the 18th Australasian Computer Science Conference, 2009, pp. 55-65.
[15] J. M. Chen, S. G. Ju, J. G. Pan, Z. W. Zu, Z. Y. Gong, "Fast Event Matching Algorithm Based on Content," Journal on Communcations. vol. 32, no. 6, June 2011.
[16] Y. Zhao, and J. Wu, "Towards Approximate Event Processing in a Large-Scale Content-Based Network," Int. Conf.Distributed Computer Systems(ICDCS), 2011 31st Int. Conf. on.,Minneapolis, MN. 2011, pp.790-799.
[17] X. Guo, J. Wei, D. Han,"Efficient Event Matching in Publish/subscribe:Based on Routing Destination and Matching History,"Networking, Architecture, and Storage(NAS'08), Int. Conf. on, Chongqing., 2008, pp. 129-136.
[18] 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.