ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

Enhanced Broker-Less Publish/Subscribe System

Monisha P Mohanan[1], Shanjana C[2]

Student, Computer Science & Engineering, Malabar Institute of Technology, Kannur, India [1]

Assistant Professor Computer Science & Engineering, Malabar Institute of Technology, Kannur, India [2]

Abstract: The basic security mechanism in publish/subscribe system is highly challenging and difficult to achieve due to the loose coupling of publishers and subscribers. This paper presents a novel approach to provide confidentiality and authentication in a broker-less content based publish/subscribe system using identity-based encryption techniques, by adapting pairing-based cryptography and also uses a lightweight encryption scheme, P-Coding, to provide confidentiality in an energy efficient way. This paper also preserves the weak subscription confidentiality in publish/subscribe system. The overall approach provides fine-grained key management and the cost for encryption, decryption and routing is in the order of subscribed attributes. It reduces the energy consumed by data encryption.

Keywords: Content-based, publish/subscribe, broker-less, identity-based encryption, security.

I. INTRODUCTION

In Publish/Subscribe System, the communication is driven by the content of the information [8] rather than the identities of producer (publisher) or consumer (subscriber). Publishers inject information into the system in the form of events, without the knowledge of relevant set of subscribers. And subscriber also specifies their interest by means of subscription without the knowledge of publishers. This is traditionally ensured by intermediate routing over a broker network. In recent system, publishers and subscribers organize themselves in a broker-less routing infrastructure. Here, publishers and subscribers contribute as peers to the maintenance of self organizing overlay structure.

Different ways to declare interest in events have produced many variants of publish/subscribe systems. The most expressive and powerful variant is content-based, here events are structured as a set of attribute/value pairs, and subscriptions are expressed as conjunction of elementary constraints over the value of one or more attributes. This is particularly useful for large scale distributed applications such as stock exchange, traffic control, news distribution etc. Publish/subscribe should provide supportive mechanisms to fulfill the basic security demands[1] of large-scale distributed applications such as authentication, confidentiality and integrity. Access control in pub/sub system means 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 exposed to the routing infrastructure. Solving these security issues in content-based pub/sub system imposes new challenges.

Existing approaches towards secure publish/subscribe systems mostly rely on the presence of a broker network. So, this paper presents a new approach to provide authentication and confidentiality in a broker-less pub/sub system. Our approach allows subscribers maintain credentials according to their subscriptions and publisher associates each encrypted event with a set of credentials. The private keys assigned to the subscribers and publishers are labeled with the credentials. Here Identity-based encryption mechanisms is adapted 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. And also preserve the weak subscription confidentiality. Furthermore, we use P-Coding, a lightweight encryption scheme to provide event confidentiality in an energy efficient way. P-Coding [2] incurs minimal energy consumption compared to other encryption schemes. The basic idea of P-Coding is to perform permutation encryption on coded messages.

II. LITERATURE SURVEY

J. Bethencourt, A. Sahai, and B. Waters [6], they have presented in their work a system for the complex access control strategy on encrypted data that is called as Ciphertext-Policy Attribute-Based Encryption. By using this technique encrypted data is being kept secret even if the storage server is not secured. In the previous research work Attribute-Based Encryption systems used attributes to describe the encrypted data and specify the policies into user's keys, while in this system attributes are used to describe a user's credentials and a party that is encrypting the data determines the policy for who can decrypt the encrypted data.

This paper present a system for realizing complex access control on encrypted data calls Ciphertext-Policy Attribute-Based Encryption. In several distributed systems a user should only be able to access data if a user posses a certain set

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
IJARCCE
nCORETech
LBS College of Engineering, Kasaragod
Vol. 5, Special Issue 1, Feburary 2016

of credentials or attributes. Currently, the only method for enforcing such policies is to employ a trusted server to store the data and mediate access control. However, if any server storing the data is compromised, then the confidentiality of the data will be compromised. In ciphertext-policy attribute-based encryption techniques, the encrypted data can be kept confidential even if the storage server is untrusted and the methods are secure against collusion attacks. Previous Attribute-Based Encryption systems used attributes to describe the encrypted data and built policies into user's keys; while in this system attributes are used to describe a user's credentials, and a party encrypting data determines a policy for who can decrypt.

M. Ion, G. Russello, and B. Crispo[3] have presented a pub/sub system that are loosely coupled where applications interact indirectly and asynchronously. Publisher generates events that are sent to interested subscribers through a network of brokers. Subscriber specifies its interest by specifying filters that is used by the brokers for the routing of events. So, it is desirable that at any mechanism that is used for protecting the confidentiality of both the events and the filters should not require that publishers and subscribers to share their secret keys. And such a mechanism should not restrict the expressiveness of filters and it should also allow the broker to perform event filtering to route the events to the interested subscribers. Existing solutions do not fully address these issues, so here they propose a mechanism that will address all these issues.

M. Srivatsa and L. Liu[5] design of PSGuard, for secure event dissemination in pub-sub networks. The key management algorithms disassociate keys from subscriber groups and ensure that the key management cost is independent of the total number of the subscribers (N S) in the pub-sub system. To achieve this in two steps: (i) First, associate an authorization key K(f) with a subscription filter f and an encryption key K(e) with an event e. Use the encryption key K(e) to encrypt the secret attributes in an event e and the authorization key K(f) to decrypt the secret attributes in a matching event e. (ii) Use hierarchical key derivation algorithms to map the authorization keys and the encryption keys into a common key space.

The mapping ensures that a subscriber can efficiently derive an encryption key K(e) for an event e using an authorization key K(f) for the subscription filter f if and only if the event e matches the subscription filter f . J. Bacon. D. M. Eyers, J. Singh[4] introduce a system architecture comprises multiple administration domains sharing a dedicated event-broker network. They found the architecture is appropriate for many applications. They also assume a secure server per domain that manages credentials and activates roles according to policy. With access control functionality located in the client-hosting brokers, enforce RBAC on the publish/subscribe clients. In general, separating event-management functionality into a dedicated event service makes access control easier to enforce than in a peer-to-peer approach where the client and event service are collocated. The latter seems inappropriate for applications transmitting sensitive data. They have assumed content-based routing, for efficiency of communication, rather than broadcast based routing.

III. SYSTEM MODEL AND BACKGROUND

A. Content-based Publish/Subscribe

Publish/subscribe system are classified as type/topic or content/attribute based[4]. Topic based publish/subscribe involves the use of an event channel dedicated to a particular named topic/type. In Content based model, it considers the message content. A subscriber defines their interest in receiving particular events based on type and attribute value. The event space denoted by Ω , is composed of a global ordered set of d distinct attributes (A_i): $\Omega = \{A_1, A_{2,\ldots,} A_d\}$. A subscription filter f is a conjunction of predicates, i.e., $f = \{Pred_1 \wedge Pred2 \ldots \wedge Pred_j\}$. $Pred_i$ is defined as a tuple (A_i, Op_i, v_i), where Op_i denotes an operator and v_i a value. An event is matched against a subscription f, if and only if the values of attributes in the event satisfy the corresponding constraints imposed by the subscription.

B. Attacker Model

Attacker model is similar to the honest-but-curious model [7][1]. In pub/sub system, publishers and subscribers are computationally bounded and do not trust each other. Also they are considered to be honest and do not deviate from the designed protocol. i.e., they route the events according to the protocol and do not drop events or forward them in a wrong manner. Publishers, who are authorized to publish, disseminate valid events in the system. But malicious publishers may masquerade the authorized publishers and spam the network with fake and duplicate events. We do not intend to solve the digital copyright problem. Therefore we assume that authorized subscribers do not reveal the content of successfully decrypted events to other subscribers. Some subscribers are curious to discover the subscriptions of other subscribers and published events to which they are not authorized to subscribe. Similarly, some publishers are also interested to read events published in the system. And passive attackers can eavesdrop the communication and try to discover contents of events and subscriptions. Finally, we assume presence of secure channels for the distribution of keys from the key server to the publishers and subscribers.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
IJARCCE
nCORETech
LBS College of Engineering, Kasaragod
Vol. 5, Special Issue 1, Feburary 2016

C. Security Goals and Requirements

The proposed secure publish/subscribe system has two goals, i.e.,security goals and scalability goals[1][7].
Authentication of Publishers and Subscribers: In order to avoid non-eligible publications only authorized publishers should be able to publish events in the system. Similarly, subscribers should only receive those messages to which they are authorized to subscribe. i.e. only authorized entities are allowed to participate in pub/sub system.

Confidentiality and Integrity of Events: The events should only be visible to authorized subscribers and should be protected from illegal modifications. Confidentiality of Subscriptions: The authorized subscribers should receive events matching their interests without revealing their subscriptions. Subscription integrity mandates that a subscriber should receive all those events which match its subscription and which are published by authorized publishers.

The secure publish/subscribe system should scale with the number of subscribers in the system. Three aspects are important to preserve scalability: i) the number of keys to be managed and the cost of subscription should be independent of the number of subscribers in the pub/sub system, ii) the re-keying overhead should be minimized.

D. Identity-based Encryption

The existing public key encryption mechanisms[8] such as PKI allows a party to encrypt data to a particular user. Senders and receivers are strongly coupled, i.e., before a sender can encrypt a message, the receiver must generate a public/private key pair, sign its public key by a certificate authority and communicate it to the sender. PKI is inefficient for a large number of subscribers as each event needs to be encrypted with each subscriber's individual public key. Therefore, a mechanism is needed to enable any pair of users to securely communicate and verify each other's signatures without the need to exchange private or public keys. And it is very hard to provide subscription confidentiality in a broker-less publish/subscribe system, where the subscribers are arranged in an overlay network according to the containment relationship between their subscriptions.

Identity-based encryption[1][7] is a promising alternative to reduce the amount of keys to be managed. In Identity-based encryption (IBE), any valid string which uniquely identifies a user can be the public key of the user. A key server maintains a single pair of public and private master keys. The master keys are used by publisher and subscriber to encrypt and decrypt the data. Master public key can be used by the sender to encrypt and send the messages to a user with any identity, e.g., an email address. To successfully decrypt the message, a receiver needs to obtain a private key for its identity from the key server. Figure 1 shows the basic idea of using Identity-based encryption.

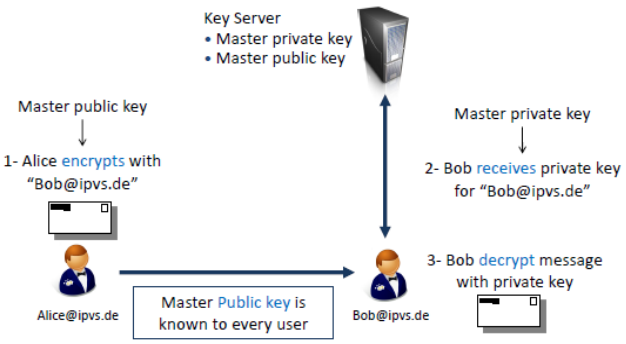

Fig.1. Identity-based Encryption

Recently pairing-based cryptography has laid the foundation of a practical implementation of Identity-based encryption[8]. In pairing-based cryptography, it establishes a mapping between two cryptographic groups. This allows reduction of one problem in one group to a different usually easier problem in another group. The mapping between cryptographic groups is achieved by means of bilinear maps, this technique is apply for establishing the basic security mechanisms in the pub/sub system.

The main properties of bilinear maps[6][1]: Let G_1 and G_2 are cyclic group of order q, where q is some large prime. A bilinear map is a function $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$ that associates a pair of elements from G_1 to elements in G_2. A bilinear map satisfies the following conditions:

1. Bilinearity. $\hat{e}(u^x, v^y) = \hat{e}(u^y, v^x) = \hat{e}(u, v)^{xy}$, for all $u, v \in G_1$ and $x, y \in Z$.
2. Non-degeneracy. $\hat{e}(u, v) \neq 1$, for all $u, v \in G_1$.
3. Computability. $\hat{e}$ can be efficiently computed.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
IJARCCE
nCORETech
LBS College of Engineering, Kasaragod
Vol. 5, Special Issue 1, Feburary 2016

E. Lightweight Encryption Scheme

P-Coding, [2] a lightweight encryption scheme. It defines permutation encryption as a special case of the classic transposition cipher. The basic idea of P-Coding is to perform permutation encryption on encoded messages. In distributed content distribution, the source may need to transmit a large volume of data, length D. In this case, the source first divides data or message into a fixed length of blocks. Each block is appended with an index value. And each and every block is encoded, and encrypted with PEF(Permutation Encryption Function). Append each encrypted blocks together and form a single string and also encrypt the combined string with PEF.

In P-Coding, before access the fitness of a permutation p, we should first decrypt generations of packets, and then decode them. The decoding process requires [2] $O(nh^2l)$ multiplications, where n is the number of generations, h is the generation size, and l is the length of a message. Use of permutation encryptions generates considerable confusion to eavesdropping adversaries. Thus, it is much more time consuming to access the fitness of p in P-coding. P-Coding is quite lightweight in computation. Recent studies demonstrate that P-Coding can help achieve lower energy consumption in MANETs. P-Coding incurs minimal energy consumption compared to other encryption schemes. And the encryption time of P-Coding is around 1/3 that of AES. The less encryption time means larger throughput. And it also means, fewer CPU cycles, and less energy consumptions.

IV. APPROACH OVERVIEW

For providing security mechanisms in pub/sub, uses the principles of identity-based encryption [1] [8], the implementation of security methods are built upon Attribute-based encryption, which is a general form of Identity-based encryption. In particular, here adapted the Ciphertext-policy Attribute-based encryption (CPABE). The authentication of publishers and subscribers as well as confidentiality of events is ensured, by adapting the pairing-based cryptography mechanisms, in terms of bilinear maps and also uses a light weight encryption scheme, P-Coding[2]. Publishers and subscribers interact with a key server by providing credentials to the key server. And in turn receives keys which fit the expressed capabilities in the credentials. Those keys are used for encrypt, decrypt and sign relevant messages in the content-based publish/subscribe system. i.e. the credential become authorized by the key server. A credential consists of two parts, first a binary string which describes the capability of a peer in publishing and receiving events, and second is proof of its identity. Publishers and subscribers pay their attention mainly at expressing the capabilities of a credential, i.e., how subscribers and publishers can create a credential[1].

The keys allotted to publishers and subscribers, and the ciphertexts are assigned credentials[7]. In particular, the Identity-based encryption ensures that a particular key can decrypt a particular ciphertext only if there is a match between the credentials of the ciphertext and the key. Publishers and subscribers maintain separate private keys for each authorized credential. The public keys are generated by a string concatenation of an credential, an epoch for key revocation, a symbol € {SUB, PUB} distinguishing publishers from subscribers. So the public keys can be easily generated by any peer without contacting the key server or other peers in the system. The published event is encrypted with the public key of all possible credentials, which authorizes a subscriber to successfully decrypt the event. The ciphertexts of the encrypted event are then signed with the private key of the publisher, as shown in Figure 2.

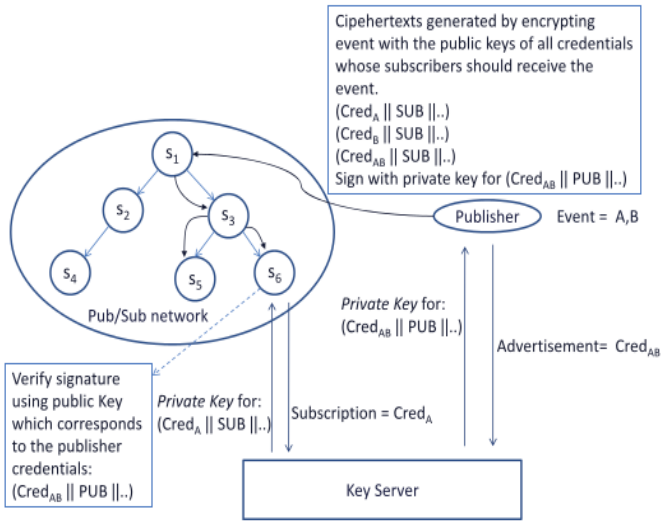

Fig. 2. Approach overview

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
IJARCCE
nCORETech
LBS College of Engineering, Kasaragod
Vol. 5, Special Issue 1, Feburary 2016

The pub/sub overlay is considered to be a virtual forest of logical trees, where each tree is associated with an attribute. Publisher sends its event to the trees associated with the attributes in the event and subscriber joins the trees corresponding to each attribute of its subscription. The subscribers are connected to the tree according to their containment relationship between credential associated with the attribute. The subscribers with coarser credentials are placed near the root of the tree and forward the events to finer credentials subscribers. For maintaining the overlay topology requires each peer should know the subscription of its parent and child peers. So, it is infeasible to provide strong subscription confidentiality. So, broker-less pub/sub system address the weak subscription confidentiality [8][1].

V. PUBLISHER/SUBSCRIBER AUTHENTICATION AND EVENT CONFIDENTIALITY

A. Security Parameters and Initialization

Let G_1 and G_2 denote the bilinear groups [1][8]of prime order q, i.e., $|G_1|= |G_2| = q$, $\hat{e}: G_1 \times G_1 \rightarrow G_2$ denotes an admissible bilinear map and g denotes a generator in G_1. Let $H_1 : \{0 , 1\}^* \rightarrow \{0, 1\}^{nu}$, $H_2 : \{0, 1\}^* \rightarrow \{0, 1\}^{nm}$, $H_3 : \{0, 1\}^* \rightarrow G_1$, and $H_4 : G_2 \rightarrow \{0, 1\}^{\log q}$. The initialization algorithm[1][7]

1. chooses $\alpha, \gamma \in Z_q$,
2. computes $g_1 = g^{\alpha}$ and $h = g^{\gamma}$,
3. chooses g_2, u', m' $\in G_1$, and
4. Selects vectors $\vec{u} = (u_i)$ and $\vec{m} = (m_i)$ of length n_u and n_m respectively with every element chosen uniformly at random from G_1.

The Master Public Key MPu is published as: $MPu = (\hat{e}, g, g_1, g_2, h, u', m', \vec{u}, \vec{m})$. This master public key is known to every peer in the system and is used for encryption and signature verification. The Master Private key MPr is (γ, g_2^{α}), and is only known to the key server. The master private key is used for generating private keys for publishers and subscribers.

IV. KEY GENERATION FOR PUBLISHERS

Before publishing an event, [1] [7] a publisher contacts the key server along with the credentials for each attribute in its advertisement. If the publisher is allowed to publish events according to its credentials, the key server will generate separate private keys for each credential. Let $Cred_{i,j}$ denotes the credential with label j for the attribute A_i.
Public Key: The public key of a publisher p for credential $Cred_{i,j}$ is generated as:
$Pu^p_{i,j} := (Cred_{i,j} \| Ai \| PUB \| Epoch)$

Private Keys: The key server will generate the corresponding private keys as follows.
$Pr^p_{i,j} := (g_2^{\alpha} (u' \prod_{k \in \tau i,j} u_k)^{\gamma i,j}, g^{\gamma i,j}) =: (Pr^p_{i,j}[1], Pr^p_{i,j}[2])$

V. KEY GENERATION FOR SUBSCRIBERS

To subscribe an event, matching its subscription, a subscriber contacts the key server and in turn receives the private keys for the credentials associated with each attribute Ai [1][8].
Public Key: The public key of a subscriber for a credential $Cred_{i,j}$ is given as:
$Pu^p_{i,j} := (Credi,j \| A_i \| SUB \| Epoch)$

Private Keys: The private keys are generated as follows.
$Pr^s_{i,j} := (g_2^{\gamma s} (u' \prod_{k \in \tau i,j} u_k)^{\gamma i,} , g^{\gamma i,j}, H_3(u' \prod_{k \in \tau i,j} u_k)^{\varphi})$
$=: (Pr^s_{i,j}[1], Pr^s_{i,j}[2], Pr^s_{i,j}[3])$

D. Publishing Events

To ensure confidentiality a publisher p encrypts an event message before publishing it. To permit subscribers to verify the authenticity and integrity of the encrypted event message, it is signed using the private keys of the publisher p. The following shows the cryptographic steps performed by the publisher p to encrypt and sign an event message M.

Encryption. When a publisher wants to publish an event message M [1], it chooses $b_i \in Z_q$ at random for each attribute A_i of the event, such that $b = \sum_{i-1}^{d} b_i$. These random values ensure that only the subscribers who have matching credentials for each of the attributes should be able to decrypt the event. The publisher generates a fixed-length random key SK for each event. The following steps are performed by the publisher to encrypt an event.

Step 1: Compute: $CT_1 = \hat{e}(g_1, g_2)^b SK$, $CT_2 = h^b$ and $CT_3 = BlockCipher(Msg \| 0^*)^{SK}$, where $Msg = (M, \{Pu^p_{i,j}\})$. The cost of asymmetric encryption generally increases with the size of the plaintext. Therefore, fixed length random key SK

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
IJARCCE
nCORETech
LBS College of Engineering, Kasaragod
Vol. 5, Special Issue 1, Feburary 2016

is encrypted using the private keys of publisher. The record Msg is encrypted with a light weight encryption scheme [2], P-Coding.

To enable the subscribers to detect the successful decryption of events, Msg is appended with a predefined number of zeros ($Msg\|0^*$). Alternatively, hash of Msg, i.e., $H_2(Msg)$, can be included in the ciphertext. Moreover, Msg includes the public keys of the publisher in addition to the event message M.

Step 2: For each attribute A_i, compute $CT_i = gb_i$. The CT_i ciphertexts along with $CT'_{i,j}$ (created in Step 3) and $Pr^s_{i,j}$ are used for the routing of encrypted events.

Step 3: For each attribute of the event, a ciphertext should be created for every credential that matches the value associated with that attribute, so that a subscriber with any of these credentials should be able to decrypt the event. For example, in case of a numeric attribute with value mapped to 0000, a ciphertext should be disseminated for the credentials 0000,000,00 and 0. For each credential $Cred_{i,j}$ that matches the value of the attribute A_i, compute
$CT_{i,j} = (u' \prod_{k \in \tau ij} u_k)^{bi}$ and
$CT'_{i,j} = H_4(\hat{e}(H_3(u' \prod_{k \in \tau ij} u_k), h^{bi}))$.

Signature: Finally, the publisher p signs the ciphertexts using its private keys. It computes $v_m = H_2(M)$ a bit string of length n_m. Let $v_m[k]$ denotes the kth bit and $\tau_m \subseteq \{1, 2, \ldots, n_m\}$ be the set of all k for which $v_m[k] = 1$. For each attribute, the credential $Cred_{i,j}$ that authorizes the publisher p to send the corresponding attribute value, p computes:
$CT^{sign}_{i,j}[1] = Pr^p_{i,j}[1](m' \prod_{k \in \tau ij} m_k)^{bi}$ and
$CT^{sign}_{i,j}[2] = Pr^p_{i,j}[2]$

E. Receiving Events

Decryption: On receiving the ciphertexts from a publisher, subscriber s tries to decrypt the event using its private keys. If the decryption is successful, the subscriber s then checks the authenticity of the decrypted event by verifying the signatures (associated with the event) using the public keys of the publisher p. The following shows the cryptographic steps performed by the subscriber s to decrypt the event message M and verify its signatures [1] [8].

Step 1: The symmetric key SK is retrieved from the ciphertext CT_1 by performing the following pairing-based cryptographic operations.

$$DT = \frac{(\prod_{i=1}^{d} \frac{\hat{e}(Pr^s_{i,\tau i}[1], CT_i)}{\hat{e}(Pr^s_{i,\tau i}[2], CT_{i,\tau i})}) CT_1}{\hat{e}(CT_2, Pr^s[4])} = SK$$

Step 2: Symmetric key SK is then used to recover $Msg = (M, \{Pu^p_{i,j}\})$ from CT_3. The successful decryption of Msg is detected by looking for predefined number of zeros appending the Msg record or verifying the hash of Msg, i.e., $H_2(Msg)$.

F. Verification

A subscriber will only accept the message if it is from an authorized publisher. [8]To check the authenticity of an event, subscribers use the master public key (MPu) and perform the following steps:

1. Compute: $VT_L = \hat{e}(\prod_{i=1}^{d} CT^{sign}_{i,j}[1], g)$, where $\prod_{i=1}^{d} CT^{sign}_{i,j}[1]$ represents the product of all received $CT^{sign}_{i,j}[1]$ ciphertexts.
2. Compute : $VT_{R1} = \prod_{i=1}^{d} \hat{e}(g_1, g_2)$.
3. Compute : $VT_{R2} = \hat{e}(\prod_{i=1}^{d}(u' \prod_{k \in \tau i,j} u_k), \prod_{i=1}^{d} CT^{sign}_{i,j}[2])$, where $\prod_{i=1}^{d}(u' \prod_{k \in \tau i,j} u_k)$ represents the product of all $Pu^p_{i,j}$ in CT_3 and $\prod_{i=1}^{d} CT^{sign}_{i,j}[2]$ is the product of all received $CT^{sign}_{i,j}[2]$ ciphertexts.
4. Compute: $VT_{R3} = \hat{e}(m' \prod_{k \in \tau m} m_k, \prod_{i=1}^{d} CT_i)$.

The received event is authentic if the following identity holds.
$VT_L = VT_{R1} \times VT_{R2} \times VT_{R3}$

VI. CONCLUSION

This work presented an approach to provide authentication and confidentiality in a broker-less content- based pub/sub system. The approach is highly scalable in terms of number of subscribers and publishers in the system and the number

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
IJARCCE
nCORETech
LBS College of Engineering, Kasaragod
Vol. 5, Special Issue 1, Feburary 2016

of keys to be maintained by them. Here, developed a mechanism to assign credentials to publishers and subscribers according to their advertisements and subscriptions. Private keys assigned to publishers and subscribers, and the ciphertexts are labeled with credentials. And, here adapted techniques from identity- based encryption and lightweight encryption, P-Coding. Identity-based encryption ensures 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 allow subscribers to verify the authenticity of received events. Lightweight encryption scheme ensures minimal energy consumption compared to other encryption scheme. Here, we expected that after the successful implementation of proposed system, the broker-less pub/sub system would show increase in performance by maintaining basic securities and energy saving.

ACKNOWLEDGMENT

We wish to thank the Department of Computer Science and Engineering also we are thankful of Project Lab for providing all resources and their valuable support.

REFERENCES

[1]. M. A. Tariq, B. Koldehofe, and K. 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]. P. Zhang, C. L.,Yixin Jiang, Y. Fan, and X. Shen, "A Lightweight Encryption Scheme for Network-Coded Mobile Ad Hoc Networks," IEEE Transactions on parallel and distributed systems, vol. 25, no. 9, September 2014
[3]. 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
[4]. J. Bacon, D. M. Eyers, J. Singh, and P.R. Pietzuch,"Access Control in Publish/Subscribe System," Proc. Second ACM Int'l Conf. Distributed Event-Based Systems(DEBS), 2008.
[5]. M. Srivatsa and L. Liu,"Secure Event Dissemination in Publish-Subscribe Networks," Proc. 27th Int'l Conf. Distributed Computing Systems, 2007
[6]. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc. IEEE Symp. Security and Privacy, 2007.
[7]. M.A. Tariq, B. Koldehofe, A. Altaweel, and K. Rothermel, "Providing Basic Security Mechanisms in Broker-Less Publish/Subscribe Systems," Proc. ACM Fourth Int'l Conf. Distributed Event-Based Systems (DEBS), 2010.
[8]. M. A. Tariq, B. Koldehofe, and K. Rothermel, "Non functional Requirements in Publish/Subscribe Systems,"

BIOGRAPHY

Monisha P Mohanan was born at Mattanur, Kerala, India. The author received Bachelor Degree in the field of Information Technology, VMKV Engineering College, Salem, Tamilnadu, India in the year 2011. She is currently pursuing her Master of Engineering in the field of Computer Science and Engineering, Malabar Institute of Technology, Kannur, Kerala. The author's areas of interest and research work involve Information Security, Network Security and Mobile Security.