Computer Networks 56 (2012) 2014–2037

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Design and implementation of a confidentiality and access control solution for publish/subscribe systems

Mihaela Ion [a], Giovanni Russello [a,b,*], Bruno Crispo [c]

[a] CREATE-NET International Research Center, Via alla Cascata 56 D, 38123 Trento, Italy
[b] Department of Computer Science, University of Auckland, Private Bag 92019, Auckland 1142, New Zealand
[c] Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

ARTICLE INFO

Article history:
Received 1 June 2011
Received in revised form 9 January 2012
Accepted 20 February 2012
Available online 28 February 2012

Keywords:
Publish/subscribe
Confidentiality
Attribute-based encryption
Encrypted search

ABSTRACT

The publish/subscribe model offers a loosely-coupled communication paradigm where applications interact indirectly and asynchronously. Publishers generate events that are sent to interested applications through a network of brokers. Subscribers express their interest by specifying filters that brokers can use for routing the events. Supporting confidentiality of messages being exchanged is still challenging. First of all, it is desirable that any scheme used for protecting the confidentiality of both the events and filters should not require publishers and subscribers to share secret keys. In fact, such a restriction is against the loose-coupling of the model. Moreover, such a scheme should not restrict the expressiveness of filters and should allow the broker to perform event filtering to route the events to the interested parties. Existing solutions do not fully address these issues. In this paper, we provide a novel scheme that supports (i) confidentiality for events and filters; (ii) allows publishers to express further constraints about who can access their events; (iii) filters that can express very complex constraints on events even if brokers are not able to access any information in clear on both events and filters; (iv) and, finally, it does not require publishers and subscribers to share keys. Furthermore, we show how we applied our scheme to a real-world e-health scenario, developed together with a hospital. We also describe the implementation of our solution in Java and the integration with an existing publish/subscribe system.

1. Introduction

The publish/subscribe (pub/sub) model is an asynchronous communication paradigm where senders, known as *publishers*, and receivers, known as *subscribers*, exchange messages in a loosely coupled manner, i.e. without establishing direct contact. The messages that publishers generate are called *events*. Publishers do not send events directly to subscribers, instead a network of interconnected brokers is responsible for delivering the events to the interested subscribers. In fact, publishers do not know who receives their events and subscribers are not aware of the source of information. In order to receive events, subscribers need to register their interest with a broker through a *filter*. When a new event is published, brokers forward it to all subscribers that expressed a filter that matches the event.

The pub/sub communication paradigm has the advantage of allowing the full decoupling of the communicating entities [1] which enables dynamic and flexible information exchange between a large number of entities. The communicating parties do not need to know each other or establish contact in order to exchange content. Moreover, if durable subscription is enabled, publishers and subscribers do not need to actively participate in the interaction at the same

* Corresponding author at: CREATE-NET International Research Center, Via alla Cascata 56 D, 38123 Trento, Italy.
E-mail addresses: mihaela.ion@create-net.org (M. Ion), g.russello@auckland.ac.nz, giovanni.russello@create-net.org (G. Russello), crispo@disi.unitn.it (B. Crispo).

1389-1286/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2012.02.013

time. If a subscriber is offline when a publisher creates an event, the broker will store the event until the subscriber becomes online and the event can be delivered.

These characteristics make the pub/sub communication model well suited for a wide range of information-driven and event-driven applications. For example, pub/sub has been proposed for information dissemination applications such as instant news delivery, stock market quotes distribution, auction bids [2], and air traffic control. Other applications of pub/sub are mobile systems [3], ubiquitous computing [4], distributed workflow management systems [5], and peer-to-peer systems [6].

Most of the research in pub/sub has been focusing on efficient routing of events [7]. However, there are many scenarios that require control over who can access the information. For example, a stock quote service could provide to paying customers information on stock prices. In this case, only paying subscribers should be able to access the events. At the same time, subscribers may wish to keep the details of their filters private from anybody spying on their interests. Another application scenario is in the medical sector where physicians are notified when certain events happen such as changes in the condition of a patient. Such information should be available to the authorised personnel only to protect the patient's privacy. Moreover, when events contain sensitive information about individuals or institutions (e.g. personal and medical data), it should be possible to enforce access control policies on the content of the events.

If the brokers are trusted, for example if they are under the direct control of the organisation using the pub/sub system, the confidentiality of the events and filters can be ensured by securing the communication between brokers, between publishers and brokers, and between brokers and subscribers. However, in many scenarios brokers cannot be considered trusted, either because a malicious employee could get access to the data (i.e. events and filters) and misuse it, or because the pub/sub system has been outsourced to another company. Outsourcing the IT infrastructure is a business model adopted more and more by companies because it reduces costs and improves the quality of services and operations. In fact, even sectors such as healthcare, initially reluctant to adopting this model, are slowly employing it [8]. Because of that, there is a need for confidentiality and access control solutions that can be applied when brokers are untrusted and could compromise the confidentiality of events and filters.

One of the main challenges that pub/sub systems are still facing is protecting the confidentiality of the exchanged information in the presence of untrusted brokers without limiting the decoupling of the paradigm. Publishers and subscribers do not establish contact so they cannot exchange keying material. Moreover, protecting the confidentiality from malicious brokers is very difficult. Brokers should be able to route events by matching them against complex filters expressed by subscribers without having access to the actual content of events and filters. Current solutions are not able to provide event and filter confidentiality, while at the same time supporting complex filters and keeping key management scalable. For example, in order to support routing based on expressive filters, [9] encrypts only certain event fields while other fields are left as cleartext so that they can be used for routing. Other solutions [10] require publishers and subscribers to share a group key which hampers the loosely coupling and scalability of the pub/sub model. Ref. [11] provides confidentiality of events and filters but the filter is restricted to equality with one keyword.

Another challenge that pub/sub systems are still facing is enforcing access control policies in the presence of untrusted brokers. Access control mechanisms usually require a trusted authority that has access to the security policies and enforces them. In fact, current access control solutions for pub/sub systems [12,13] rely on brokers to enforce the policies, and assume brokers are trusted both with the content of policies and events. Such solutions cannot be applied in an outsourced environment in which brokers are untrusted. Enforcing access control policies on the content of the events without revealing the access policy and the event content to the brokers is still an open issue.

The main contribution of this paper is to present an approach catering for confidentiality in pub/sub systems when brokers are untrusted such that: (i) it provides confidentiality of events and filters, (ii) it allows publishers to express further constraints about who can access their events, (iii) it does not require publishers and subscribers to share keys, and (iv) it allows subscribers to express filters that can define any monotonic and non-monotonic conditions. To achieve this, our solution combines attribute-based encryption and an encrypted search scheme. We show how to apply our solution to provide confidentiality of messages exchanged in a real-world e-health application. We then implement our scheme in Java and integrate it with an existing publish/subscribe system. We measure the overhead introduced for encrypting events and filters and performing encrypted match.

The rest of the paper is structured as follows: Section 2 introduces the pub/sub communication model and provides an example of an application where pub/sub confidentiality and access control are required. Section 3 describes the problem of confidentiality and the properties achieved by our solution. Section 4 presents related work and shows that none of the current solutions supports all the properties identified by us. Section 5 introduces the relevant encryption mechanisms used by our scheme. Sections 7 and 6 give the details of our solution. Section 8 revises the application example described in Section 2 implemented with our approach. Section 9 provides the security analysis of our scheme. Section 10 describes the implementation and the integration with an existing pub/sub system. Section 11 provides the performance evaluation and Section 12 concludes the paper.

2. The publish/subscribe communication paradigm

Several pub/sub implementations that differ in the granularity used in the definition of the filters have been proposed in the literature. The most simple one is *topic-based*, in which subscribers subscribe to a topic identified by a *keyword* [14]. A topic-based scheme is similar to the notion of group communication. When subscribing to a