International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

Protecting Broker-Less Subscribe / Publish Systems through Identity-Based Encryption

Ms. S. Veena
Department OF Computer Science and Engineering, Aurora's Technological and Research Institute, Hyderabad
Mrs. T. Padmaja
Ass.Professor, Department OF Computer Science and Engineering, Aurora's Technological and Research Institute, Hyderabad

Abstract :
The provisioning of basic security mechanisms such as authentication and confidentiality is highly challenging in a content based publish/subscribe system. Authentication of publishers and subscribers is difficult to achieve due to the loose coupling of publishers and subscribers. Likewise, confidentiality of events and subscriptions conflicts with content-based routing. This paper presents a novel approach to provide confidentiality and authentications in a broker-less content-based publish/subscribe system. The authentication of publishers and subscribers as well as confidentiality of events is ensured, by adapting the pairing-based cryptography mechanisms, to the needs of a publish/subscribe system. Furthermore, an algorithm to cluster subscribers according to their subscriptions preserves a weak notion of subscription confidentiality.

In addition to our previous work, this paper contributes 1) use of searchable encryption to enable efficient routing of encrypted events, 2) multi credential routing a new event dissemination strategy to strengthen the weak subscription confidentiality, and 3) thorough analysis of different attacks on subscription confidentiality. The overall approach provides fine-grained key management and the cost for encryption, decryption, and routing is in the order of subscribed attributes. Moreover, the evaluations show that providing security is affordable w.r.t. 1) throughput of the proposed cryptographic primitives, and 2) delays incurred during the construction of the publish/subscribe overlay and the event dissemination.

I INTRODUCTION

The publish/subscribe (pub/sub) communication paradigm has gained high popularity because of its inherent decoupling of publishers from subscribers in terms of time, space, and synchronization. Publishers inject information into the pub/sub system, and subscribers specify the events of interest by means of subscriptions. Published events are routed to their relevant subscribers, without the publishers knowing the relevant set of subscribers, or vice versa. This decoupling is traditionally ensured by intermediate routing over a broker network. In more recent systems, publishers and subscribers organize themselves in a broker-less routing infrastructure, forming an event forwarding overlay. Content-based pub/sub is the variant that provides the most expressive subscription model, where subscriptions define restrictions on the message content. Its expressiveness and asynchronous nature is particularly useful for large-scale distributed applications such as news distribution, stock exchange, environmental monitoring, traffic control, and public sensing. Not surprisingly, pub/sub needs to provide supportive mechanisms to fulfill the basic security demands of these applications such as access control and confidentiality. Access control in the context of 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. Moreover, the content of events should not be exposed to the routing infrastructure and a subscriber should receive all relevant events without revealing its subscription to the system. Solving these security issues in a content-based pub/sub 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, publishers must maintain the public keys of all interested subscribers to encrypt events. Subscribers must know the public keys of all relevant publishers to verify the authenticity of the received events. Furthermore, traditional mechanisms to provide confidentiality by encrypting the whole event message conflict with the content-based routing paradigm. Hence, new mechanisms are needed to route encrypted events to subscribers without knowing their subscriptions and to allow subscribers and publishers authenticate each other without knowing each other. Building on our results of, this paper presents a new approach to provide authentication and confidentiality in a broker-less pub/sub system. Our approach allows subscribers to maintain credentials according to their subscriptions.

Private keys assigned to the subscribers are labeled with the credentials. A publisher associates each encrypted event with a set of credentials. We adapted identity-based encryption (IBE) mechanisms) 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. Furthermore, we address the issue of subscription confidentiality in the presence of semantic clustering of subscribers. A weaker notion of subscription confidentiality is defined and a secure overlay maintenance protocol is designed to preserve the weak subscription confidentiality.

II SYSTEM ANALYSIS

Existing Systems:

- Content-based publish/subscribe is the variant which pro-vides the most expressive subscription model, where subscriptions de ne restrictions on the message content. Its expressiveness and asynchronous nature is particularly useful for large-scale distributed applications with high-volume data streams.
- Access control in the context of publish/subscribe system means that only authenticated publishers are allowed to disseminate events in the network and only those events are delivered to authorized subscribers. Similarly, the content of events should not be exposed to the routing infrastructure and a subscriber should receive all relevant events without revealing its subscription to the system. These security issues are not trivial to solve in a content-based pubish/subscribe system and pose new challenges.

Disadvantages Of Existing System:

- It is very hard to provide subscription condentiality in a broker-less publish/subscribe system, where the subscribers are arranged in an overlay network according to the containment relationship between their subscriptions. In this case, regardless of the cryptographic primitives used, the maximum level of attainable condentiality is very limited.
- The limitation arises from the fact that a parent can decrypt every event it forwarded to its children. Therefore, mechanisms are needed to provide a weaker notion of condentiality.
- Do not intend to solve the digital copyright problem.

Proposed System:

- In this paper, we present a new approach to provide authentication and condentiality in a broker-less publish/subscribe system.
- Our approach allows subscribers to maintain credentials according to their subscriptions.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

Private keys assigned to the subscribers are labelled with the credentials.

- A publisher associates each encrypted event with a set of credentials. We adapted identity based encryption mechanisms.

Advantages Of Proposed System:

- To ensure that a particular subscriber can decrypt an event only if there is match between the credentials associated with the event and the key.

To allow subscribers to verify the authenticity of received events. Furthermore, we address the issue of subscription condentiality in the presence of semantic clustering of subscribers. A weaker notion of subscription condentiality is dened and a secure connection protocol is designed to preserve the weak subscription condentiality. Finally, the evaluations demonstrate the viability of the proposed security mechanisms.

III SYSTEM IMPLEMENTATION

Modules:

- Publishers Module
- Subscribers Module
- Key Generation for Publishers/Subscribers
- Publishing Events

Publishers Module

In the first module, we develop Publishers Module. Publisher's module is designed to have options to Register publishers and then login to their account. Publishers have option to publish about the topics. And if publishers tries to publish any duplicate content which is already published by them, then it will block and notify them too. The module is also designed with access of publisher to see the request of Subscribers. Only if the publisher accepts the subscriber, then only the subscriber can view the published contents. In publishers and subscribers module, both the entities are computationally bounded and do not trust each other.

Moreover, all the peers (publishers or subscribers) participating in the pub/sub overlay network are honest and do not deviate from the designed protocol. Likewise, authorized publishers only disseminate valid events in the system. However, malicious publishers may masquerade the authorized publishers and spam the overlay network with fake and duplicate events. We do not intend to solve the digital copyright problem; therefore, authorized subscribers do not reveal the content of successfully decrypted events to other subscribers.

Subscribers Module

In this module, we develop the Subscribers Module. Subscribers are, however, curious to discover the subscriptions of other subscribers and published events to which they are not authorized to subscribe. Similarly, curious publishers may be interested to read events published in the system. Furthermore, passive attackers outside the pub/sub overlay network can eavesdrop the communication and try to discover content of events and subscriptions.

Key Generation for Publishers/Subscribers

Before starting to publish events, 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. The key server will generate the corresponding private keys.

Similarly, to receive events matching its subscription, a subscriber should contact the key server and receive the private keys for the credentials associated with each attribute

Publishing Events

Encryption: When a publisher wants to publish an event message M, it random for each attribute Ai of the event. These random values ensure that only the subscribers who have matching credentials for each of the attributes should be able to decrypt the event.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

Furthermore, the publisher generates a fixed-length random key SK for each event. The cost of asymmetric encryption generally increases with the size of the plaintext. Therefore, only a fixed-length random key SK is encrypted using the private keys of publisher. The record Msg is encrypted with a symmetric encryption algorithm such as AES or Triple DES, using key SK.

Decryption: On receiving the cipher texts, a subscriber tries to decrypt them using its private keys. The cipher texts for each attribute are strictly ordered according to the containment relation between their associated credentials; therefore, a subscriber only tries to decrypt the cipher text whose position coincides with the position of its credential in the containment hierarchy of the corresponding attribute. The position of a credential can be easily determined by calculating its length.

IV SYSTEM MODEL

(A) Content-Based Publish/Subscribe

For the routing of events from publishers to the relevant subscribers, we use the content-based data model. The event space, denoted by Ω, is composed of a global ordered set of d distinct attributes (A_i): $\Omega = \{A_1, A_2, \ldots, A_d\}$ Each attribute $\mathbf{A_i}$ is characterized by a unique name, its data type, and its domain. The data type can be any ordered type such as integer, floating point, and character strings. The domain describes the range $[L_i, U_i]$ of possible attribute values. A subscription filter f is a conjunction of predicates, i.e., $f = \{Pred_1 \wedge Pred_2 \cdots \wedge Pred_j\}$. $Pred_i$ is defined as a tuple (A_i, Op_i, v_i), where Opi denotes an operator and vi a value. The operator Opi typically includes equality and range operations for numeric attributes and prefix/suffix operations for strings. An event consists of attributes and associated values. An event is matched against a subscription f if the values of attributes in the event satisfy the corresponding constraints imposed by the subscription.

(B) Attacker Model

It is similar to the commonly used honestbut- curious model. We have two entities in the system: publishers and subscribers. Both the entities are computationally bounded and do not trust each other. Moreover, all the peers (publishers or subscribers) participating in the pub/sub overlay network are honest and do not deviate from the designed protocol. Likewise, authorized publishers only disseminate valid events in the system.

However, malicious publishers may masquerade the authorized publishers and spam the overlay network with fake and duplicate events. We do not intend to solve the digital copyright problem; therefore, authorized subscribers do not reveal the content of successfully decrypted events to other subscribers. Subscribers are, however, curious to discover the subscriptions of other subscribers and published events to which they are not authorized to subscribe. Similarly, curious publishers may be interested to read events published in the system. Furthermore, passive attackers outside the pub/sub overlay network can eavesdrop the communication and try to discover content of events and subscriptions.

(C) Security Goals and Requirements

Authentication. To avoid noneligible 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.

Confidentiality. In a broker-less environment, two aspects of confidentiality are of interest: 1) the events are only visible to authorized subscribers and are protected from illegal modifications, and 2) the subscriptions of subscribers are confidential and unforgeable.

Scalability. The secure pub/sub system should scale with the number of subscribers in the system. Three aspects are important to preserve scalability: 1) the number of keys to be managed and the cost of subscription should be independent of the number of subscribers in the system, 2) the key server and subscribers should maintain small and constant numbers of keys per subscription, and 3) the overhead

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

because of rekeying should be minimized without compromising the fine-grained access control.

(D) Identity-Based Encryption

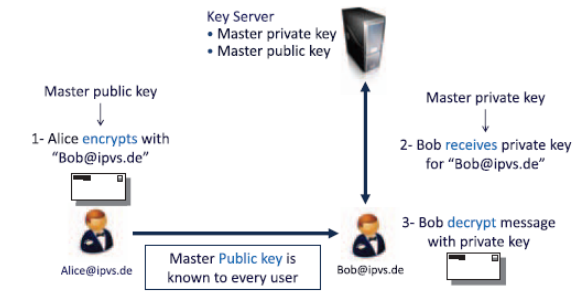

Fig. 1. Identity-based encryption.

While a traditional PKI infrastructure requires to maintain for each publisher or subscriber a private/public key pair which has to be known between communicating entities to encrypt and decrypt messages, identity-based encryption provides a promising alternative to reduce the amount of keys to be managed. In identity-based encryption, 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 public key can be used by the sender to encrypt and send the messages to a user with any identity, for example, an e-mail address. To successfully decrypt the message, a receiver needs to obtain a private key for its identity from the key server. Fig. 1 shows the basic idea of using identity-based encryption..

V PUBLISHER/SUBSCRIBER AUTHENTICATION AND EVENT CONFIDENTIALITY

(A) Security Parameters and Initialization

Let G1 and G2 denote the bilinear groups of prime order q, i.e., $|\mathbb{G}_1| = |\mathbb{G}_2| = q$, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ denote an admissible bilinear map, and g denote a generator in G1. Moreover, let $H_1 : \{0,1\}^* \to \{0,1\}^{n_u}$, $H_2 : \{0,1\}^* \to \{0,1\}^{n_m}$, $H_3 : \{0,1\}^* \to \mathbb{G}_1$, and $H_4 : \mathbb{G}_2 \to \{0,1\}^{\log q}$ designate collusion esistant cryptographic hash functions. The initialization algorithm

1. chooses $\alpha, \varphi \in \mathbb{Z}_q$,
2. computes $g_1 = g^\alpha$ and $h = g^\varphi$,
3. chooses $g_2, u', m' \in \mathbb{G}_1$, 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 $\mathbb{G}_1$.

(B) Key Generation for Publishers/Subscribers

Publisher keys. Before starting to publish events, 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 Credi;j denote the credential with label j for the attribute Ai, for example, CredTemp;0 denotes credential 0 of attribute Temp. The public key of a publisher p for credential Credi;j is generated as

$$Pu^p_{i,j} := (Cred_{i,j} \parallel A_i \parallel PUB \parallel Epoch).$$

The key server will generate the corresponding private keys as follows: For each credential Credi;j and a publisher p, let $v_p = H_1(Pu^p_{i,j})$ be a bit string of length nu and let $v_p[k]$ denote the kth bit. Let $\Gamma_{i,j} \subseteq \{1, 2, \ldots, n_u\}$ nug be the set of all k for which $v_p[k] = 1.$. The key server chooses $\gamma_{i,j} \in \mathbb{Z}_q$ at random and computes

$$Pr^p_{i,j} := \left(g_2^\alpha \left(u' \prod_{k \in \Gamma_{i,j}} u_k \right)^{\gamma_{i,j}}, g^{\gamma_{i,j}} \right) =: (Pr^p_{i,j}[1], Pr^p_{i,j}[2]).$$

Subscriber keys. Similarly, to receive events matching its subscription, a subscriber should contact the key server and receive the private keys for the credentials associated with each attribute Ai. In case of subscribers, the public key for a credential Credi;j is given as

$$Pu^s_{i,j} := (Cred_{i,j} \parallel A_i \parallel SUB \parallel Epoch).$$

A different symbol SUB is used to differentiate the keys used for the verification of valid events from the ones used to provide event confidentiality. The private keys are generated as follows: The key server chooses $\gamma_s \in \mathbb{Z}_q$ at random. The same γ_s is used for all credentials associated with a subscription. For each credential $Cred_{i,j}$, it calculates $\Gamma_{i,j}$ similar to the publisher's case, chooses $\gamma_{i,j} \in \mathbb{Z}_q$ and computes

$$Pr^s_{i,j} := \left(g_2^{\gamma_s} \left(u' \prod_{k \in \Gamma_{i,j}} u_k \right)^{\gamma_{i,j}}, g^{\gamma_{i,j}}, H_3 \left(u' \prod_{k \in \Gamma_{i,j}} u_k \right)^{\varphi} \right)$$
$$=: (Pr^s_{i,j}[1], Pr^s_{i,j}[2], Pr^s_{i,j}[3]).$$

VI CONCLUSION

In this paper, we have presented a new 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 of keys maintained by them. In particular, we have developed mechanisms to assign credentials to publishers and subscribers according to their subscriptions and advertisements. Private keys assigned to publishers and subscribers, and the cipher texts are labeled with credentials.

We adapted techniques from identity based encryption 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 its private keys and 2) to allow subscribers to verify the authenticity of received events. Furthermore, we developed a secure overlay maintenance protocol and proposed two event dissemination strategies to preserve the weak subscription confidentiality in the presence of semantic clustering of subscribers. The evaluations demonstrate the viability of the proposed security mechanisms and analyze attacks on subscription confidentiality.

REFERENCES

[[1] D. Akhawe, P. Saxena, and D. Song. Privilege separation in HTML5 applications. In Proceedings of the 21st Usenix Security Symposium, Bellevue, WA, Aug. 2012.

[2] A. Arasu, S. Blanas, K. Eguro, R. Kaushik, D. Kossmann, R. Ramamurthy, and R. Venkatesan. Orthogonal security with Cipherbase. In Proceedings of the 6th Biennial Conference on Innovative Data Systems Research (CIDR), Asilomar, CA, Jan. 2013.

[3] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In Proceedings of the 13th Annual Network and Distributed System Security Symposium, San Diego, CA, Feb. 2006.

[4] S. Bajaj and R. Sion. TrustedDB: a trusted hardware based database with privacy and data confidentiality. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, pages 205–216, Athens, Greece, June 2011.

[5] A. Barth, C. Jackson, and J. C. Mitchell. Securing frame communication in browsers. In Proceedings of the 17th Usenix Security Symposium, San Jose, CA, July–Aug. 2008.

[6] A. Barth, J. Caballero, and D. Song. Secure content sniffing for web browsers, or how to stop papers from reviewing themselves. In Proceedings of the 30th IEEE Symposium on Security and Privacy, Oakland, CA, May 2009.

[7] F. Beato, M. Kohlweiss, and K. Wouters. Scramble! your social network data. In Proceedings of the 11th Privacy Enhancing Technologies Symposium, Waterloo, Canada, July 2011.

[8] D. Benjamin. Adapting Kerberos for a browserbased environment. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Sept. 2013.

[9] D. Borelli. The name Edward Snowden should be sending shivers up CEO spines. Forbes, Sept. 2013. http://www.forbes.com/sites/ ealspin/2013/09/03/the-name-edwardsnowden-should-be-sending-shivers-upceo- spines/.

[10] A. Chen. GCreep: Google engineer stalked teens, spied on chats. Gawker, Sept. 2010. http:// gawker.com/5637234/.