Secure Publish and Subscribe Systems with Efficient Revocation

Sana Belguith, Shujie Cui, Muhammad Rizwan Asghar, Giovanni Russello
Cyber Security Foundry, The University of Auckland
Auckland, New Zealand
{sbel452,scui379}@aucklanduni.ac.nz,{r.asghar,g.russello}@auckland.ac.nz

ABSTRACT

User revocation is one of the main security issues in publish and subscribe (pub/sub) systems. Indeed, to ensure data confidentiality, the system should be able to remove malicious subscribers without affecting the functionalities and decoupling of authorised subscribers and publishers. To revoke a user, there are solutions, but existing schemes inevitably introduce high computation and communication overheads, which can ultimately affect the system capabilities.

In this paper, we propose a novel revocation technique for pub/sub systems that can efficiently remove compromised subscribers without requiring regeneration and redistribution of new keys as well as re-encryption of existing data with those keys. Our proposed solution is such that a subscriber's interest is not revealed to curious brokers and published data can only be accessed by the authorised subscribers. Finally, the proposed protocol is secure against the collusion attacks between brokers and revoked subscribers.

KEYWORDS

Secure Pub/Sub, Publications' Confidentiality, Subscribers' Privacy, Collusion Resistance

ACM Reference Format:
Sana Belguith, Shujie Cui, Muhammad Rizwan Asghar, Giovanni Russello. 2018. Secure Publish and Subscribe Systems with Efficient Revocation. In *SAC 2018: SAC 2018: Symposium on Applied Computing , April 9–13, 2018, Pau, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3167132.3167176

1 INTRODUCTION

Publish and Subscribe (pub/sub) systems enable dissemination of data contents from data producers to interested users. Data produced by these producers (*a.k.a. publishers*) is routed to the interested users (*a.k.a. subscribers*) through a network of dedicated servers known as *brokers*. The pub/sub paradigm allows publishers to send data (*a.k.a. publications*) to subscribers who subscribe their interests (*a.k.a. subscriptions*) in a loosely-coupled manner, *i.e.,* without establishing a direct contact.

In pub/sub systems, publishers might not be aware of subscribers and their interests. To receive the publications, the subscribers define their interests as *filters* and register them with the broker. Each publication is composed of a piece of content and its *tags* that summarises the content. When a new publication is published, the broker forwards it to the registered subscribers whose filter matches the publication's tags.

Thanks to its characteristics, pub/sub systems have been widely used in several fields, such as e-health information systems to share health records between involved organisations, *i.e.,* hospitals, doctors and pharmacies, stock exchange services to publish available trades to consumers, news services. For instance, news agencies use the news service to sell their content to customers, and many others [20].

Despite its benefits, pub/sub systems raise serious privacy and security concerns due to the involvement of untrusted brokers that perform matching and routing of publications through multi-party distributed communication systems. First, publications' confidentiality is considered as one of the main security issues in pub/sub systems. Even if the publication content is protected, publication tags might reveal sensitive information about the publication. On one hand, brokers should have access to publication tags for performing match in order to route the publications to the subscribers; on the other hand, providing such access to brokers can result in confidentiality breach. Second, the privacy of subscribers is another challenge in pub/sub systems. In fact, brokers are considered as curious entities that might harvest data about subscribers and their interests. Without subscribers' interest, brokers might not be able to perform the matching. The main issue is to allow brokers to perform matching but without revealing interests in the cleartext. Third, a compromised or revoked subscriber can collude with a broker in order to identify the interest of registered subscribers. Specifically, a compromised subscriber can collude with the broker and registers her subscriptions in cleartext. Thus, even if other subscriptions are encrypted, the broker can still infer the content by checking if they match against the same publications with the subscriptions in cleartext. Therefore, interests should be protected from both brokers and subscribers.

Beyond ensuring confidentiality and privacy, as a multi-user distributed system, a pub/sub system brings the issue of user revocation in an efficient manner. Obviously, malicious subscribers should be removed from the system so that they are unable to receive publications. A naive solution could be

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SAC 2018, April 9–13, 2018, Pau, France
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5191-1/18/04…$15.00
https://doi.org/10.1145/3167132.3167176

to regenerate new keys and redistribute them to authorised subscribers and publishers as well as re-encrypt all the existing data with those keys. However, this solution presents several drawbacks. First, when a large number of subscriptions need to be updated, the key update can take longer and incur huge computation and communication overheads. Second, the re-encryption of the publications also introduces a huge computation overhead, which can ultimately affect the system capabilities. Third, the reliance on a third party such as proxy to re-encrypt data can be a bottleneck as this entity must remain active and uncorrupted throughout the lifetime of the system. In short, such a naive approach might work on small scale, but it is not suitable for medium to large enterprises, where subscribers might be joining or leaving more frequently.

To mitigate this, some research solutions propose to allow direct re-encryption at the broker side [19, 20, 23]. However, the revocation becomes ineffective when the revoked subscribers collude with the broker. Therefore, the challenge is to define a smooth revocation mechanism which does not require updating the secret keys of the remaining users and the re-encryption of publications.

In this paper, we introduce a secure pub/sub system achieving efficient revocation without requiring regeneration and redistribution of keys, and re-encryption of existing data with those keys. The novelty of our solution lies in the fact that an organisation can revoke a compromised or leaving subscriber by simply sending an instruction to the broker. Considering a large number of subscribers joining or leaving the organisation, this solution is scalable because the revocation operation does not require involvement or interaction with other subscribers.

Our proposal is multi-fold. First, we protect user subscriptions from curious brokers as well as other subscribers. Second, publications are only accessed by the authorised subscribers due to the usage of a symmetric encryption scheme. Third, we present an efficient revocation mechanism without affecting existing subscribers or the data. Finally, the proposed protocol is secure against the collusion attacks between brokers and revoked subscribers.

The remainder of this paper is organised as follows. Section 2 describes a pub/sub scenario and defines security requirements to be fulfilled. Then, in Section 3, we first define the system model and the threat model considered in our proposal. Next, we explain the framework design and present our detailed architecture. Afterwards, a rigorous security analysis is presented in Section 4. In Section 5, we review the related work. Finally, we conclude in Section 6 and provide some directions for future research.

2 BACKGROUND AND REQUIREMENTS

2.1 Publish and Subscribe Systems

Pub/sub systems are usually categorised into two main types: Topic-based pub/sub systems and Content-based pub/sub systems.

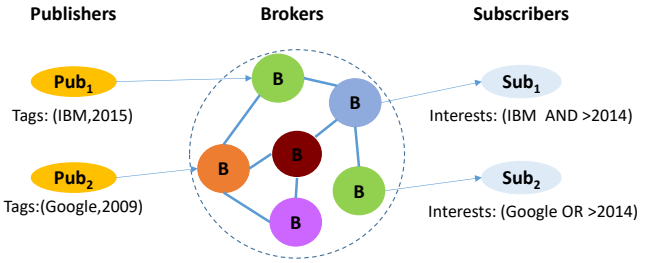

Figure 1: Content-based pub/sub system: The publisher forwards publications and tags to the brokers. The brokers perform the matching and forward the publications to the subscribers whose interest match the tags.

In topic-based pub/sub systems, subscribers choose one or more topics of their interest, which are already predefined. Hence, publications tagged with topics are forwarded to all the interested subscribers. In content-based pub/sub systems, the subscribers define their interest based on a set of predicates that are constraints over attributes. The publications contain tags that summarise the contents in a set of constraints over attributes (*cf.* Figure 1). As such, publications are forwarded to a subscriber if there is a match between her interests and the publications' tags. Content-based pub/sub systems are more flexible than topic-based pub/sub and allow subscribers to specify expressive subscriptions based on the actual content of the event [20]. Thanks to their expressiveness, in this paper, we focus on the use of a content-based pub/sub system scenario.

In this section, we begin by introducing a motivating scenario for pub/sub systems. Then, we define the security requirements to be fulfilled by the proposed solution. Finally, we provide the major research challenges that we address.

2.2 Motivating Scenario

As e-health systems are witnessing increased popularity [12, 13, 20], several health organisations are using these systems in order to share medical data in an efficient way. Indeed, an e-health information system can benefit from content-based pub/sub services to share patients' Electronic Health Records (EHR) between health organisations such as doctors, hospitals, clinics, pharmacists, *etc.* (*cf.* Figure 2).

A publisher who can be a doctor from hospital A, shares an EHR with other doctors from hospital B, a pharmacist, or a medical laboratory. In this case, the shared EHR file contains personal information about the patient such as her identity, her address, nature of the test, and the content of the file. This information must be routed to various health organisations, possibly geographically separated and in independent administrative domains, where the patient can be moved when her condition stabilises or where tests have to be performed or analysed.

It is noteworthy that the preservation of the publication's confidentiality is not the only security concern. It is crucial

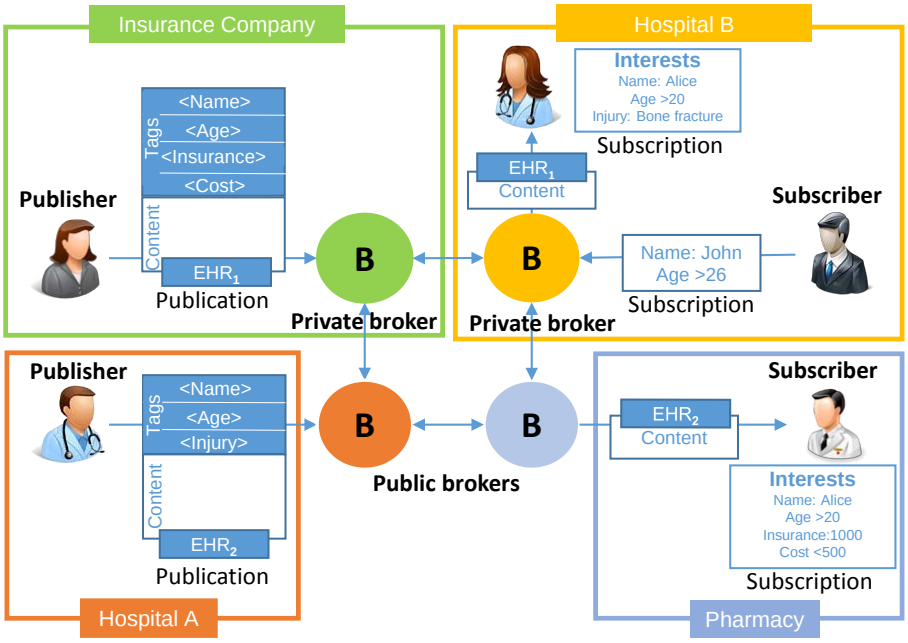

Figure 2: E-health pub/sub service: Several brokers from different domains are connected. The publishers forward patient EHRs to the employed public/private brokers. Via the routing of multiple brokers, the EHRs could be forwarded to the intended subscribers, including those in other domains.

to ensure the confidentiality of the publication tags (name, address of the patient, nature of the test, *etc.*) which are highly sensitive information.

In addition, subscriptions are also highly sensitive information as they can reveal which patient is treated by which clinic or for which type of ailment. For instance, the system should not reveal any private information related to a doctor as well as her patients' data. That is, the disclosure of such information may be used to produce targeted advertisement related to the health condition of the patients, or to run statistical surveys.

Beyond ensuring the confidentiality of publications, tags and subscriptions, an efficient revocation should be addressed. For instance, if one of those medical actors leave the system, then she should not be able to access the published data. Suppose that a doctor has left the hospital; thus, she should not access patients' records anymore.

The design of the solution is motivated by providing the support of both robustness and efficiency while fulfilling the following properties:

Subscription Privacy. Interests submitted to brokers by the subscribers should not reveal information about the subscribers.

Tag Privacy. Data tags associated with the publications should not reveal any information about the data content.

Publication Confidentiality. The published data can only be accessed only by authorised subscribers.

Efficient Revocation. A revoked subscriber should not be able to access the publications after she leaves the system.

2.3 Security Requirements and Challenges

Based on the aforementioned e-health scenario, we define the security requirements to provide a secure pub/sub service as follows:

R1. The content of publications should be concealed from the brokers.

R2. The broker should be able to check if the tags of publications match against subscriber's interests without knowing the content of them.

R3. Published data should be accessed only by authorised subscribers whose interests match the publication tags. In other words, a revoked subscriber should not be able to access the publications that were published after she leaves the system.

R4. When any malicious behaviour is detected, the malicious subscriber should be revoked without affecting the other subscribers.

3 PROPOSED SOLUTION

In this work, we propose a novel architecture for ensuring efficient revocation of subscribers in pub/sub systems. Our proposed technique does not require regeneration and redistribution of keys when a subscriber is revoked. Furthermore, our solution does not need leak any information of subscribers to

the publishers. In addition, our proposal introduces a secure design against collusion between brokers and subscribers.

To meet R1, the contents of publications are encrypted before sending to the brokers. To achieve R2, our solution is built on top of Searchable Encryption (SE) schemes, where a broker could check if there is a match between tags and interests while they are encrypted. The challenge is how to guarantee R3 without limiting the decoupling of the paradigm, *i.e.,* without letting the publishers know any information of both the revoked and authorised subscribers. We address R3 and R4 by designing a new scheme (explained later), which is based on secret sharing.

In this section, we first define the system model and threat model considered in our proposal. Then, we present a general overview of our proposed solution. Finally, we provide a detailed description of the different phases used in the protocol.

3.1 System Model

We consider a pub/sub system involving the following entities:

Publisher (Pub): The publishers encrypt *publications* before publishing to the brokers. In addition, it generates *tags* related to the publication to be matched by the broker against the subscribers' interests.

Subscriber (Sub): Each subscriber declares her interests by defining subscription conditions as a *filter*, such that the subscriber only receives the data whose tags satisfy the filter.

Broker (B): The broker is responsible for filtrating and delivering matched publications to interested subscribers.

Trusted Authority (TA): The trusted authority is responsible for managing the subscribers' secret keys and revoking subscribers.

3.2 Threat Model

In pub/sub systems, we consider that the TA is fully trusted in the system and the channels between the TA and the publishers/subscribers are secure. In our system, we consider the following threat model:

Malicious Sub. A malicious Sub might try to access publications without authorisation. A malicious Sub could be one who is revoked and should not be able to access published data. In addition, a Sub is considered as malicious if she allows an unauthorised entity to access published data using her own secret keys or if her secret keys have been stolen by an adversary.

Honest but Curious Broker. The brokers are semi-trusted (honest-but-curious) in the system. They obey the protocol to evaluate the deployed filters but they are curious about the content of publications and filters. Furthermore, a broker may collude with any subscriber to allow her access to the data without being authorised. In our setting, we consider that at least three types of brokers should be present to perform the publish services. Moreover, we assume that at most two brokers can collude together to allow a revoked subscriber access to the publication.

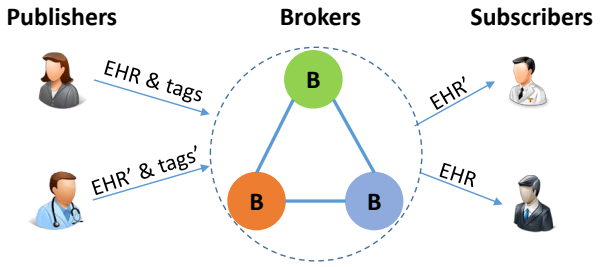

Figure 3: An overview of our system: Three types of brokers in different domains are connected into a virtual cluster. The publishers in these domains send publications to the cluster. The three types of brokers in the cluster perform the matching and routing separately, and finally only the subscribers whose filters match the tags could get the publications.

3.3 Approach Overview

In this proposal, we aim to not only protect the publications and filters but also ensure efficient revocation of Subs. As mentioned, an SE scheme is used in our system to ensure the confidentiality of tags and filters while allowing the brokers to perform the matching. To ensure efficient revocation, we design a very efficient and secure pub/sub protocol that can protect the publications from revoked Subs without any key regeneration or publication re-encryption operations. Basically, in our system, three different types of brokers running in different domains are combined into a cluster. We separate the filter matching, publication routing and forwarding functionalities and deploy them in the three non-colluding brokers. As illustrated in Figure 3, the Pubs in these domains send publications to the cluster. The three types of brokers in the cluster process the publications and forward them to the Subs whose filters match the tags.

In our setting, each Sub has three secret keys, and each of them is shared with a broker. When the Sub is revoked, the TA forwards her SN to the brokers so that the subscriber's three keys will be removed by the three types of brokers. To protect the publication from revoked subscribers and malicious brokers, the Pubs encrypt their publications with three nonces. Then, each nonce is sent to one of the brokers to ensure that each broker is unable to decrypt the received publications. When the publication tags match the filter of an authorised Sub, each broker re-encrypts the publication under the Sub's shared secret key in an efficient manner without learning the content of the publication. Therefore, only the authorised subscriber can decrypt the publications using her three secret keys. We assume that at most two brokers can collude together or can collude with a revoked subscriber. When at least of one broker processes the encrypted publication properly, the revoked subscribers are unable to decrypt the publication.

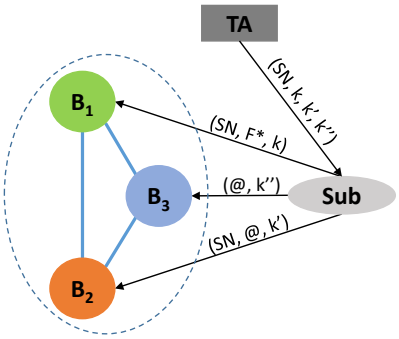

Figure 4: Subscription phase: When subscribing, the Sub first gets three keys k, k', k'' and a Serial Number (SN) from the TA. Each Sub is identified with a unique SN. The filter F is encrypted with SE before sending to the brokers. The matched publication will be sent to the Sub based on its registered address @. The Sub shares one secret key with each broker.

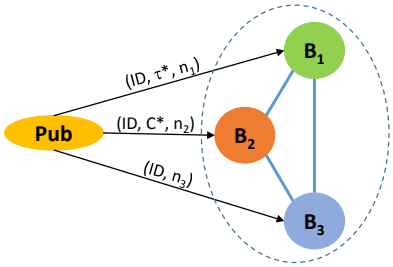

Figure 5: Publication phase: Each publication is identified with a unique ID. The tags τ is encrypted into τ^* with SE, and the content of the publication C is encrypted with three nonces n_1, n_2 and n_3 by using the one-time pad.

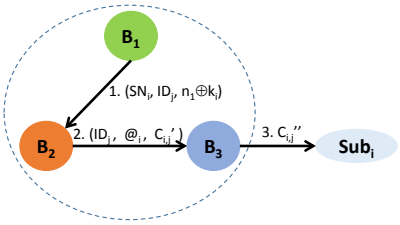

Figure 6: Matching, routing and forwarding phase: B_1 matches τ^* against F^*, and forwards all the matched (SN_i, ID_j) pairs as well as $k_i \oplus n_1$ to B_2. For each matched pair, B_2 replace SN_i with its corresponding $@_i$, generates $C'_{i,j}$ by computing $C^*_j \oplus (n_1 \oplus k_i) \oplus k'_i \oplus n_2$ and sends $(ID_j, @_i, C'_{i,j})$ to B_3. Finally, B_3 computes $C''_{i,j} \leftarrow C'_{i,j} \oplus k''_i \oplus n_3$ and forwards it to Sub_i.

3.4 Detailed Architecture

We use B_1, B_2 and B_3 to represent the three types of brokers that provides matching, routing and forwarding functionalities, respectively.

Our proposed solution consists of 7 different procedures detailed as follows:

- **Subscribe**: Each Sub_i is identified with a unique serial number (SN_i) generated by TA and an address ($@_i$) using which the subscriber could receive publications from the brokers. Let F_i be the filter of Sub_i, she registers with the system as follows:
  - Sub_i first requests her secret keys k_i, k'_i, k''_i and serial number SN_i from the TA.
  - Second, Sub_i encrypts F_i into F^*_i using a SE.
  - Finally, she sends (SN_i, F^*_i, k_i), (SN_i, k'_i) and $(SN_i, @_i, k''_i)$ to the broker B_1, B_2 and B_3, respectively (*cf.* Figure 4).
- **Publish**: Let $P_j = (ID_j, \tau_j, C_j)$ be a publication to be published, where ID_j is the identifier, τ_j represents a set of tags, and C_j is the content of the publication, respectively.
  The Pub encrypts P_j as below:
  - The Pub first uniformly and independently chooses three nonces $n_1 \xleftarrow{\$} \{0,1\}^{|C_j|}$, $n_2 \xleftarrow{\$} \{0,1\}^{|C_j|}$, $n_3 \xleftarrow{\$} \{0,1\}^{|C_j|}$, where $|C_j|$ is the bit length of C_j;
  - Second, the Pub encrypts C_j by computing $C^*_j \leftarrow C_j \oplus n_1 \oplus n_2 \oplus n_3$;
  - Meanwhile, the Pub encrypts the tags with the SE scheme, *i.e.,* $\tau^*_j \leftarrow Enc_{SE}(\tau_j)$;
  - Finally, as depicted in Figure 5, the Pub forwards (ID_j, τ^*_j, n_1), (ID_j, C^*_j, n_2) and (ID_j, n_3) to B_1, B_2 and B_3, respectively.
- **Matching on** B_1: B_1 gets (SN_i, F^*_i, k_i) from Sub_i and (ID_j, τ^*_j, n_1) from a Pub. For each Sub_i's filter and each publication P_j,
  - B_1 first checks if the encrypted tags τ^*_j match against the encrypted filter F^*_i based on the SE scheme.
  - If yes, B_1 computes $n_1 \oplus k_i$.
  - Finally, it forwards $(SN_i, ID_j, n_1 \oplus k_i)$ to B_2 for each matched filter and publication pair (*cf.* Figure 6, Step 1).
- **Routing on** B_2: B_2 gets $(SN_i, @_i, k'_i)$ from Sub_i and (ID_j, C^*_j, n_2) of P_j. For each message $(SN_i, ID_j, n_1 \oplus k_i)$ get from B_1,
  - B_2 first gets $(@_i, k'_i)$ indexed by SN_i, and (C^*_j, n_2) indexed by ID_j.
  - Then, it computes $C'_{i,j} \leftarrow C^*_j \oplus (k_i \oplus n_1) \oplus n_2 \oplus k'_i$
  - Finally, it forwards $(ID_j, @_i, C'_{i,j})$ to B_3 (*cf.* Figure 6, Step 2).
- **Forwarding on** B_3: B_3 gets $(@_i, k''_i)$ from Sub_i and (ID_j, n_3) of P_j. For each message $(ID_j, @_i, C'_j)$ get from B_2,
  - B_3 first gets n_3 indexed by ID_j and k''_i indexed by $@_i$.
  - Second, it computes $C''_{i,j} \leftarrow C'_{i,j} \oplus n_3 \oplus k''_i$.
  - Finally, $C''_{i,j}$ is sent to Sub_i (*cf.* Figure 6, Step 3).

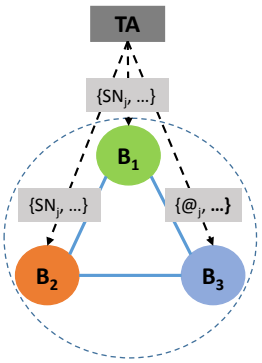

Figure 7: Subscriber revocation: he TA sends a list of SNs and @s of the revoked Subs to the brokers. The three types brokers will remove the revoked Subs information from their storage.

- **Decryption on** Sub_i: Formally, after the processing by the three types of brokers, the re-encrypted publication is $C''_{i,j} = C_j \oplus k_i \oplus k'_i \oplus k''_i$. Sub_i knows k_i, k'_i and k''_i, and it can get C_j by computing $C''_{i,j} \oplus k_i \oplus k'_i \oplus k''_i$.
- **Revocation**: If Sub_i is revoked, the TA sends its SN_i to B_1 and B_2, and sends its $@_i$ to B_3 (*cf.* Figure 7). All the information of Sub_i, including her filter and keys, will be removed from B_1, B_2 and B_3. Then, the publication will never be encrypted under its secret keys. Thus, Sub_i is unable to recover the content of any publication anymore. Note that, to achieve fine-grained access control, the TA could generate attribute-specific keys to Subs. If the Sub is just revoked the access to the publications with certain attributes. The brokers just need to remove her keys related to these attributes.

4 SECURITY ANALYSIS

The security of the encrypted filters and tags is based on the SE scheme used in our system. The proposed protocol is designed to ensure the confidentiality of the publications against both unauthorised subscribers and curious brokers. In this section, we prove the security of encrypted publications, based on two realistic threat models, as defined in Section 3.2.

As described in Section 3.4, by using the one-time pad encryption, the publisher first encrypts the publication content C_j:

$$C^*_j \leftarrow C_j \oplus n_1 \oplus n_2 \oplus n_3 \quad (1)$$

and forwards it to B_2. After the matching on B_1, B_2 gets $n_1 \oplus k_i$, using which B_2 could generate

$$C'_{i,j} \leftarrow C_j \oplus n_3 \oplus k_i \oplus k'_i \quad (2)$$

Then, $C'_{i,j}$ is sent to B_3 and converted into

$$C''_{i,j} \leftarrow C_j \oplus k_i \oplus k'_i \oplus k''_i \quad (3)$$

In all the phases, the C_j is always protected with three parameters, *i.e.,* (n_1, n_2, n_3) in (1), (n_3, k_i, k'_i) in (2) and (k_i, k'_i, k''_i) in (3). However, each broker only knows one of them, indicating the publication is concealed from all the brokers. Furthermore, only Sub_i who has k_i, k'_i and k''_i could recover the publication.

Even if any two brokers collude together, the publication is still concealed by the third parameter. For instance, if B_1 colludes with B_2, they can only get $C_j \oplus n_3$ by computing $C^*_j \oplus n_1 \oplus n_2$, where n_3 is only known to B_3. If B_1 colludes with B_3, they can only get $C_j \oplus k'_i$ by computing $C'_{i,j} \oplus n_3 \oplus k_i$, where k'_i is only known to B_2. Similarly, if B_2 and B_3 collude together, they could only get $C_j \oplus n_1$ and $n_1 \oplus k_i$. Without either n_1 or k_i from B_1, they are unable to learn C_j. Thus, the publication content is protected from the brokers when three of them do not collude together.

Due to the third parameter, when a revoked Sub colludes with any two brokers, they are also unable to recover the publication. When a Sub is revoked, as specified in the protocol, her secret keys should be removed from the three brokers. Even if she colludes any two brokers, they could keep her keys and re-encrypt the publication with them or share the nonces with the revoked Sub. However, without the proper re-encryption process on the third broker or the third nonce, they are still unable to decrypt the publication.

5 RELATED WORK

Confidentiality in pub/sub systems has been widely studied [2, 7, 10, 20, 23]. Several works have been proposed to ensure publications' confidentiality based on encryption techniques [1, 11, 14, 17, 25, 26].

Cheng *et al.* [9] have proposed a protocol to preserve confidentiality of publications. In this solution, the broker is only responsible for forwarding subscriptions and publications to the publishers and subscribers, respectively. Basically, the subscribers send their identifiers, interests and public keys to the broker, and then the broker forwards them to the publishers. When there are related publications generated, the publishers encrypt the publications using the subscriber's public key and forwards them to the broker. Finally, the broker sends the publications to the intended subscribers. Thus, the publishers store the subscriptions and know which subscribers will get their publications. Although this solution protects the publications against the brokers, it does not fulfil the non-coupled requirement of pub/sub systems.

Tariq *et al.* [24] have introduced a secure pub/sub protocol based on Identity-Based Encryption (IBE) schemes [5]. In their approach, publishers and subscribers interact with a key server. They provide credentials to the key server and in turn receive keys that fit the expressed capabilities in the credentials. Due to the loose coupling between publishers and subscribers, a publisher does not know the set of relevant subscribers in the system. Therefore, a publication is encrypted with the public key of all possible credentials, which authorises a subscriber to successfully decrypt the publication. Afterwards, the publisher signs the generated ciphertexts of the encrypted publications with the private key of the publisher. Using IBE, a subscriber can decrypt the ciphertext only if there is a match between the credentials

of the ciphertext and her secret key. Obviously, a subscriber verifies the authenticity of the received ciphertext and then decrypts it using her private key.

Pal *et al.* have proposed PS3 [22] to enhance the subscribers' privacy and the publications' confidentiality in pub/sub systems. Their proposal relies on the combination of the Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [4] and the Predicate-Based Encryption (PBE) [6, 15]. First, the subscriber interacts with a certification authority to request her secret keys and a related certificate. Then, she sends the certificate along with her interests to a Predicate-Based Encryption Token Server (PBE-TS), which verifies her certificate and then returns the corresponding PBE token. The publisher generates a publication and a Global Unique Identifier (GUI) used to identify the publication. Then, she encrypts the publication content using CP-ABE, and the GUI using PBE. Afterwards, the publisher sends the encrypted publication content to a storage server and the encrypted GUI to the broker. The broker forwards the encrypted GUI to the registered subscribers. Thus, a subscriber whose PBE token can decrypt the publication GUI, requests the publication from the storage server. This approach could ensure the confidentiality of publications effectively. However, the matching operation is performed on subscribers and all the GUIs have to be sent to the subscribers no matter whether they are interested in them or not. Thus, it increases the computation and communication overheads on subscribers.

There are some works that considered the revocation issue in pub/sub systems [19, 23]. For instance, an efficient revocation becomes more and more hard to offer with the involvement of a huge number of users [3, 16, 18].

To ensure revocation in pub/sub systems, Onica *et al.* [19] have proposed a key update mechanism for pub/sub systems. This proposal is based on using the Asymmetric Scalar-product Preserving Encryption (ASPE) scheme to encrypt publications before storing them on the broker side. When a revocation occurs, the subscribers' secret keys are regenerated and redistributed. In addition, to ensure that the publications are accessed by authorised users, the authors introduce an *in-broker re-encryption*. Indeed, the brokers encrypt all the remaining publications using the generated secret keys. However, the revocation mechanism brings additional computation overheads at the broker end. For instance, the revocation of a subscriber requires the re-encryption of the entire published contents. Moreover, the revocation can fail if the revoked subscriber colludes with the broker. Thus, the broker does not re-encrypt the publications and sends them to the revoked user who can decrypt their contents using her secret key.

A secure proxy re-encryption scheme based on Ring-LWE (RLWE) key switching approach [8] has been proposed by Polyakov *et al.* [23] to ensure publications' confidentiality in pub/sub systems. In this proposal, the publisher encrypts publications using her public key then forwards it to a broker that acts as a proxy re-encryption server. In this proposal, the broker re-encrypts the received publications using a re-encryption token received from a trusted third party. Afterwards, the broker forwards the publication to the intended subscribers who use their secret keys to decrypt it. The broker is responsible for revoking subscribers. Indeed, it re-encrypts publications only for non-revoked users. Although this solution preserves publications' confidentiality, it assumes that the broker is fully trusted in performing the revocation procedure.

Using Palliar cryptosystem [21], Naveel *et al.* [17] have constructed a privacy-preserving context-based pub/sub system allowing brokers to perform matching without learning the content of the publications and subscriptions. Moreover, by using ABE-based Group Key Management (AB-GKM) approach, the publishers are able to enforce fine-grained access control policy to the publications. Basically, the system employs a trusted key manager to generate the keys for subscribers and publishers. For each publication, the publisher controls subscribers' access by constructing an access structure T and drives the encryption key from the 'public information'. Only the subscribers whose attributes satisfy T could derive the decryption key from the same 'public information'. If one subscriber is revoked, the key manager just needs to broadcast the updated the public information to publishers and remained subscribers. That is, no key re-generation and data re-encryption are required in this solution. However, both the publisher and subscriber need to perform expensive ABE encryption and decryption operations, which limits the practical usage of the solution.

None of the above existing solution addresses an efficient revocation of subscribers in pub/sub systems. Moreover, state-of-the-art solutions assume a weak threat model, where subscribers and brokers can not collude.

6 CONCLUSIONS AND FUTURE DIRECTIONS

Pub/sub systems have been widely used in different fields. However, this paradigm introduces security and privacy challenges. In this paper, we design a pub/sub protocol that not only preserves publications' confidentiality and subscribers' privacy, but also introduces an efficient revocation mechanism that prevents the revoked subscriber from receiving publications without updating subscribers' secret keys or limiting the decoupling aspect of the paradigm.

As for future work, we plan to further improve the proposed solution by exploring new mechanisms that can ensure the resistance against the collusion between the three types of brokers and malicious subscribers. Moreover, we will implement a prototype of our system and test its performance.

REFERENCES

[1] Muhammad Rizwan Asghar, Ashish Gehani, Bruno Crispo, and Giovanni Russello. 2014. PIDGIN: Privacy-preserving interest and content sharing in opportunistic networks. In *Proceedings of the 9th ACM symposium on information, computer and communications security*. ACM, 135–146.

[2] Raphaël Barazzutti, Pascal Felber, Hugues Mercier, Emanuel Onica, and Etienne Riviere. 2017. Efficient and confidentiality-preserving content-based publish/subscribe with prefiltering. *IEEE Transactions on Dependable and Secure Computing* 14, 3 (2017), 308–325.

[3] Sana Belguith, Nesrine Kaaniche, Abderrazak Jemai, Maryline Laurent, and Rabah Attia. 2016. PAbAC: A Privacy preserving Attribute based framework for fine grained Access Control in clouds. In *SECRYPT 2016: 13th International Conference on Security and Cryptography*, Vol. 4. Scitepress, 133–146.

[4] John Bethencourt, Amit Sahai, and Brent Waters. 2007. Ciphertext-policy attribute-based encryption. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE, 321–334.

[5] Dan Boneh and Matt Franklin. 2001. Identity-based encryption from the Weil pairing. In *Advances in Cryptology – CRYPTO 2001*. Springer, 213–229.

[6] Dan Boneh and Brent Waters. 2007. Conjunctive, subset, and range queries on encrypted data. *Theory of cryptography* (2007), 535–554.

[7] Cristian Borcea, Yuriy Polyakov, Kurt Rohloff, Gerard Ryan, et al. 2017. PICADOR: End-to-end encrypted Publish–Subscribe information distribution with proxy re-encryption. *Future Generation Computer Systems* 71 (2017), 177–191.

[8] Zvika Brakerski and Vinod Vaikuntanathan. 2011. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *Annual cryptology conference*. Springer, 505–524.

[9] Tracy Yingying Cheng, Wei Gao, Xiaohua Jia, Jianfei He, and Shucheng Liu. 2016. Privacy-preserving publish/subscribe service in untrusted third-party platform. In *Communications (ICC), 2016 IEEE International Conference on*. IEEE, 1–6.

[10] Giovanni Di Crescenzo, Jim Burns, Brian Coan, John Schultz, Jonathan Stanton, Simon Tsang, and Rebecca N Wright. 2013. Efficient and private three-party publish/subscribe. In *International Conference on Network and System Security*. Springer, 278–292.

[11] Abebe Abeshu Diro, Naveen Chilamkurti, and Neeraj Kumar. 2017. Lightweight Cybersecurity Schemes Using Elliptic Curve Cryptography in Publish-Subscribe fog Computing. *Mobile Networks and Applications* (2017), 1–11.

[12] Christian Esposito and Mario Ciampi. 2015. On Security in Publish/Subscribe Services: A Survey. *IEEE Communications Surveys and Tutorials* 17, 2 (2015), 966–997.

[13] Mihaela Ion, Giovanni Russello, and Bruno Crispo. 2010. Supporting Publication and Subscription Confidentiality in Pub/Sub Networks. In *SecureComm*. Springer, 272–289.

[14] Mihaela Ion, Giovanni Russello, and Bruno Crispo. 2012. Design and implementation of a confidentiality and access control solution for publish/subscribe systems. *Computer networks* 56, 7 (2012), 2014–2037.

[15] Vincenzo Iovino and Giuseppe Persiano. 2008. Hidden-vector encryption with groups of prime order. In *International Conference on Pairing-Based Cryptography*. Springer, 75–88.

[16] Tao Jiang, Xiaofeng Chen, and Jianfeng Ma. 2016. Public integrity auditing for shared dynamic cloud data with group user revocation. *IEEE Trans. Comput.* 65, 8 (2016), 2363–2373.

[17] Mohamed Nabeel, Stefan Appel, Elisa Bertino, and Alejandro Buchmann. 2013. Privacy preserving context aware publish subscribe systems. In *International Conference on Network and System Security*. Springer, 465–478.

[18] Mohamed Nabeel, Ning Shang, and Elisa Bertino. 2012. Efficient privacy preserving content based publish subscribe systems. In *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*. ACM, 133–144.

[19] Emanuel Onica, Pascal Felber, Hugues Mercier, and Etienne Rivière. 2015. Efficient key updates through subscription re-encryption for privacy-preserving publish/subscribe. In *Proceedings of the 16th Annual Middleware Conference*. ACM, 25–36.

[20] Emanuel Onica, Pascal Felber, Hugues Mercier, and Etienne Rivière. 2016. Confidentiality-preserving publish/subscribe: A survey. *ACM Computing Surveys (CSUR)* 49, 2 (2016), 27.

[21] Pascal Paillier et al. 1999. Public-key cryptosystems based on composite degree residuosity classes. In *Eurocrypt*, Vol. 99. Springer, 223–238.

[22] Partha Pal, Greg Lauer, Joud Khoury, Nick Hoff, and Joe Loyall. 2012. P3S: A privacy preserving publish-subscribe middleware. In *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer, 476–495.

[23] Yuriy Polyakov, Kurt Rohloff, Gyana Sahu, and Vinod Vaikuntanthan. 2017. Fast Proxy Re-Encryption for Publish/Subscribe Systems. *IACR Cryptology ePrint Archive* 2017 (2017), 410.

[24] Muhammad Adnan Tariq, Boris Koldehofe, and Kurt Rothermel. 2014. Securing broker-less publish/subscribe systems using identity-based encryption. *IEEE transactions on parallel and distributed systems* 25, 2 (2014), 518–528.

[25] Yuan Tian, Biao Song, Mohammad Mehedi Hassan, and Eui-nam Huh. 2013. An efficient privacy preserving Pub–Sub system for ubiquitous computing. *International Journal of Ad Hoc and Ubiquitous Computing* 12, 1 (2013), 23–33.

[26] Kan Yang, Kuan Zhang, Xiaohua Jia, M Anwar Hasan, and Xuemin Sherman Shen. 2017. Privacy-preserving attribute-keyword based data publish-subscribe service on cloud platforms. *Information Sciences* 387 (2017), 116–131.