Scientific Journal Impact Factor (SJIF): 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

SECURE PUBLISH/SUBSCRIBE SYSTEMS WITH BROKER-LESS CONCEPT

Jiya Joy[1], Alga Baby[2]
[1,2]*Department of Computer Science, IIET, Nellikuzhi,*

Abstract- 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. This paper presents a novel approach to provide confidentiality and authentication in a broker-less content-based publish/subscribe system. This paper, also include a model that the secure friend discovery process as a generalized privacy preserving interest and profile matching problem. We identify a new security threat arising from existing secure friend discovery protocols, coined as runaway attack, which can introduce a serious unfairness issue. Here adapted pailler's encryption mechanisms to ensure that a particular subscriber can match their interests (hobbies) with the publisher. Based on the interest match they can send friends request each other. After accepting the friends request they become friends and can send images, videos, and messages. Also we can send hidden messages inside the video. To thwart this new threat, we introduce a novel blind vector transformation technique, which could hide the correlation between the original vector and transformed results. Based on this, we propose our privacy- preserving and fairness-aware interest and profile matching protocol, which allows one party to match its interest with the profile of another.

Keywords- Cryptography,profilematching-protocol, publish/subscribe,brokerless

I. INTRODUCTION

The publish/subscribe 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 [1]. In more recent systems, publishers and subscribers organize themselves in a broker-less routing infrastructure, forming an event forwarding overlay [2]. 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 fulfil 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.

We also present a novel Privacy Preserving and Fairness-aware Friend Matching Protocol. In the designed protocol, a successful matching between subscriber and publisher happens in case that the interests of both of the participants could match the profiles of the others. In other words, no one can learn any extra information from the protocol unless another participant is exactly what he is looking for and vice versa. Our work is motivated from a simple observation that if two vectors match, they will still match no matter whether they are transformed in the same way (e.g., add or remove a randomly generated vector) or shuffled with the same order. To achieve this goal, we introduce a novel *Blind Vector Transformation* technique, under which each participant con- tributes a part of the vector transformation while any single one of the parties cannot recover the original vectors from the final transformation result. Therefore, with blind vector transformation, we could enable a party to match its interests with another's profile but, at the same time, to keep the interests as well as the profiles private. Further, to thwart runaway attack, we introduce a lightweight verifier checking technique, which enables the verifier to check the matching at the minimized overhead and prevent any participant from launching the runaway attack.

II. RELATED WORKS

For providing security mechanisms in pub/sub, we leverage the principles of identity-based encryption to support many-to-many interactions between subscribers and publishers. Although we subsequently demonstrate the implementation of our security methods in terms of a concrete variant called attribute-based encryption, it is important to remark that our approach also benefits from other identity-based encryption schemes**.**

In this approach, publishers and subscribers interact with a key server. They provide credentials to the key server and in turn receive keys which fit the expressed capabilities in the credentials. Subsequently, those keys can be used to encrypt, decrypt, and sign relevant messages in the content based pub/sub system, i.e., the credential becomes authorized by the key server. A credential consists of two parts: 1) a binary string which describes the capability of a peer in publishing and receiving events, and 2) a proof of its identity. The latter is used for authentication against the key server and verification whether the capabilities match the identity of the peer. While this can happen in a variety of ways, for example, relying on challenge response, hardware support, and so on, we pay attention mainly at expressing the capabilities of a credential, i.e., how subscribers and publishers can create a credential. This process needs to account for the many possibilities to partition the set of events expressed by an advertisement or subscription and exploits overlaps in subscriptions and publications. Subsequently, we use the term credential only for referring to the capability string of a credential. The keys assigned to publishers and subscribers, and the cipher texts, are labelled with credentials.

The keys assigned to publishers and subscribers, and the cipher texts, are labelled with credentials. In particular, the identity-based encryption ensures that a particular key can decrypt a particular cipher text only if there is a match between the credentials of the cipher text and the key. Publishers and subscribers maintain separate private keys for each authorized credential.

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 [3] 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.

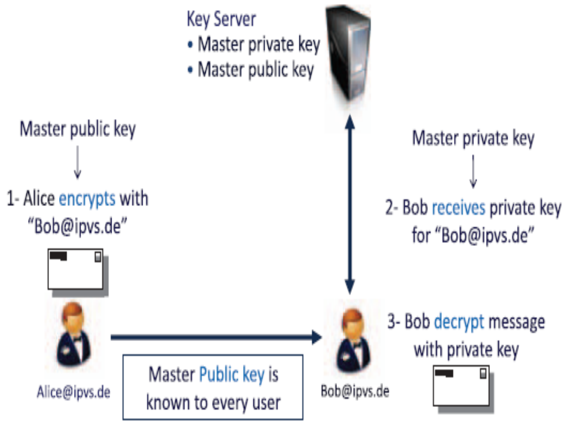


Fig. 1. Identity-based encryption.

We want to stress here that although identity-based encryption at the first glance appears like a highly centralized solution, its properties are ideal for highly distributed applications. A sender needs to know only a single master public key to communicate with any identity. Similarly, a receiver only obtains private keys for its own identities. Furthermore, an instance of central key server can be easily replicated within the network. Finally, a key server maintains only a single pair of master keys and, therefore, can be realized as a smart card, provided to each Participant of the system. Although identity-based encryption has been proposed some time ago, only recently pairing-based cryptography (PBC) has laid the foundation of practical implementation of identity-based encryption. Pairing-based cryptography establishes a mapping between two cryptographic groups by means of bilinear maps. This allows the reduction of one problem in one group to a different usually easier problem in another group.

III. PROPOSED WORK

Publisher and subscriber can interact iff there is a successful match between their interests.Both users can handle the role as subscriber or publisher. . In the designed protocol, a successful matching between subscriber and publisher happens in case that the interests of both of the participants could match the profiles of the others. In other words, no one can learn any extra information from the protocol unless another participant is exactly what he is looking for and vice versa. Our work is motivated from a simple observation that if two vectors match, they will still match no matter whether they are transformed in the same way (e.g., add or remove a randomly generated vector) or shuffled with the same order. To achieve this goal, we introduce a novel *Blind Vector Transformation* technique, under which each participant con- tributes a part of the vector transformation while any single one of the parties cannot recover the original vectors from the final transformation result. Therefore, with blind vector transformation, we could enable a party to match its interests with another's profile but, at the same time, to keep the interests as well as the profiles private.Matching protocol is based on paillier's homomorphic encryption.

Homomorphic encryption is a form of encryption that allows computations to be carried out on cipher text, thus generating an encrypted result which, when decrypted, matches the result of operations performed on the plaintext. This is sometimes a desirable feature in modern communication system architectures. Homomorphic encryption would allow the chaining together of different services without exposing the data to each of those services. Homomorphic encryption schemes are malleable by design. This enables their use in cloud computing environment for ensuring the confidentiality of processed data. In addition the homomorphic property of various cryptosystems can be used to create many other secure systems, for example secure voting systems, collision-resistant hash functions, private information retrieval schemes, and many more.

There are several partially homomorphic cryptosystems, and also a number of fully homomorphic cryptosystems. Although a cryptosystem which is unintentionally malleable can be subject to attacks on this basis, if treated carefully homomorphism can also be used to perform computations securely. A cryptosystem that supports *arbitrary computation* on ciphertexts is known as fully homomorphic encryption (FHE) and is far more powerful. Such a scheme enables the construction of programs for any desirable functionality, which can be run on encrypted inputs to produce an encryption of the result. Since such a program need never decrypts its inputs, it can be run by an untrusted party without revealing its inputs and internal state. The existence of an efficient and fully homomorphic cryptosystem would have great practical implications in the outsourcing of private computations, for instance, in the context of cloud computing.

Users can interact each other by sending messages, images and video. Message encryption is done by pailler's algorithm. Image encryption is done by XOR based algorithm. Video encryption is done by identity based encryption. Also we can hide messages inside the video clips. The main characteristics of the proposed method are:

- For the first time, we separate the user's interest from its profile.
- We introduce a novel blind vector transformation technique, which could hide the correlation between the original vector and the transformed result. Based on it, we propose the privacy-preserving and fairness-aware friend matching protocol, which enables one party to match its interest with the profile of another, and vice versa, without revealing their real interest.
- We introduce a novel lightweight verifier checking approach to thwart runaway attack and thus achieve the fairness of two participants.
- We implement our protocols in real experiments.
- After making friends publisher can send message, images and video clips to the subscriber. Here both party can act as publisher and subscriber.
- Also we can hide private messages inside the video clip.
- Message encryption is done by pailler's algorithm. Image encryption is done by XOR based algorithm. Video encryption is done by identity based encryption.

Main advantages of the proposed system are:

- Highly scalable
- Preserve weak subscription confidentiality
- Use multi credential routing
- Efficient analysis of attacks
- Enable efficient routing

PAILLIER’S ENCRYPTION

The Paillier cryptosystem, named after and invented by Pascal Paillier in 1999, is a probabilistic asymmetric algorithm for public key cryptography. This is a partially homomorphic cryptosystem. The scheme is an additive homomorphic cryptosystem; this means that, given only the public-key and the encryption of m1 and m2, one can compute the encryption of m1+m2.

- **Key Generation**
  - Constructs two randomly generated prime numbers p and q.
  - Compute n=p*q and lambda=lcm (p-1, q-1).
  - Select random integer g.
  - Publishes MK: (n,g)
  - SK :(g, n, lambda)

- **Encryption**
  - Let m be the message to be encrypted where mЄZ_n^{*}
  - Select a random r where r ЄZ_n^{*}.
  - Compute ciphertext as:

$$c=g^{m} \bullet r^{n} \bmod n^{2}$$

- **Decryption**
  - Let c be the ciphertext to decrypt where cЄ$Z^{*}_{n^{2}}$
  - Compute plaintext message as:

$$m= L (c^{lambda} \bmod n^{2}) * u \bmod n,$$
$$\text{where } u = (L(g^{lambda} \bmod n^{2}))^{(-1)} \bmod n.$$

IV. RESULTS

The important screenshots are:

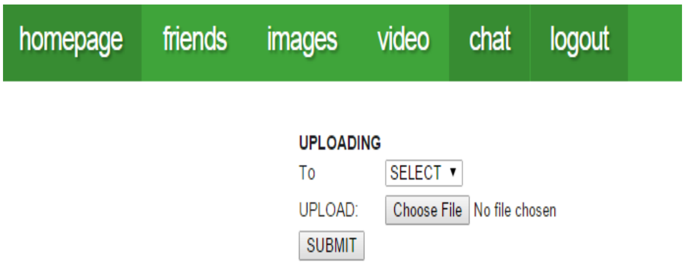


Figure 2: Upload Image

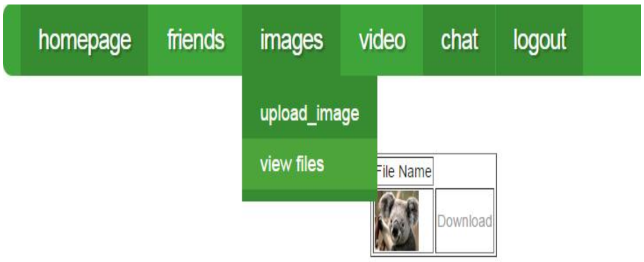
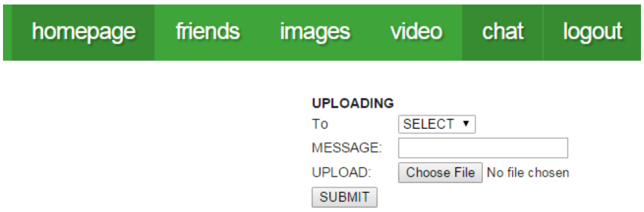
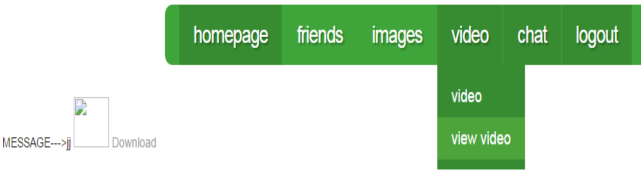


Figure 3: Download Image



UPLOADING
To SELECT ▾
MESSAGE:
UPLOAD: Choose File No file chosen
SUBMIT

Figure 4: Upload video



MESSAGE--->jj Download

Figure 5: Download video with secret message

V. CONCLUSION

Publisher and subscriber can interact if there is a successful match between their interests.Both users can handle the role as subscriber or publisher. . In the designed protocol, a successful matching between subscriber and publisher happens in case that the interests of both of the participants could match the profiles of the others. Users can interact each other by sending messages, images and video. Message encryption is done by pailler's algorithm. Image encryption is done by XOR based algorithm. Video encryption is done by identity based encryption. Also we can hide messages inside the video clips.

REFERENCES

[1] Anceaume, M. Gradinariu, A.K. Datta, G. Simon, and A. Virgillito, "A Semantic Overlay for Self- Peer-to-Peer Publish/ Subscribe," Proc. 26th IEEE Int'l Conf. Distributed Computing Systems (ICDCS), 2006.

[2] J. Bacon, D.M. Eyers, J. Singh, and P.R. Pietzuch, "Access Control\ in Publish/Subscribe Systems," Proc. Second ACM Int'l Conf. Distributed Event-Based Systems (DEBS), 2008.

[3] W.C. Barker and E.B. Barker, "SP 800-67 Rev. 1. Recommendation for the TripleData Encryption Algorithm (TDEA) Block Cipher," technical report, Nat'l Inst. Of Standards & Technology, 2012.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc. IEEE Symp.Security and Privacy, 2007.

[5] D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," Proc. Int'l Conf. Theory and Applications of Cryptographic Techniques on Advances in Cryptology (EUROCRYPT), 2004.

[6] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Int'l Cryptology Conf. Advances in Cryptology,

[7] S. Choi, G. Ghinita, and E. Bertino, "A Privacy-Enhancing Content-Based Publish/Subscribe System Using Scalar Product Preserving Transformations," Proc. 21st Int'l Conf. Database and Expert Systems Applications: Part I, 2010.

[8] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. ACM 13th Conf. Computer and Comm. Security (CCS), 2006.

[9] 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 (Secur- eComm), 2010.

[10] H.-A. Jacobsen, A.K.Y. Cheung, G. Li, B. Maniymaran, V. Muthusamy, an kazemzadeh, "The PADRES Publish/ Subscribe System," Principles and Applications of Distributed Event-Based Systems. IGI Global, 2010.