# [DEMO] ABE to the Rescue: Efficient Encrypted Communications for Disaster Management

Hongmiao Yu*, Jiachen Chen†, K. K. Ramakrishnan*
* Department of Computer Science and Engineering, University of California Riverside, Riverside, USA
† WINLAB, Rutgers University, North Brunswick, NJ, USA.
hyu125@ucr.edu, jiachen@winlab.rutgers.edu, kk@cs.ucr.edu

***Abstract*—Efficient and secure message dissemination plays an important role during a disaster environment. Name-based publish/subscribe systems, especially role-based names, using principles of Information-Centricity provide an efficient framework for communications among first responders. However, a challenge is maintaining confidentiality during communication. We have developed an encryption framework that leverages graph-based naming systems which provides role-based communication among first responders. Our framework is built on top of the dynamic role-based names and can be implemented using attribute-based encryption (ABE) or public key encryption (PKE). In this demo, we show the operations of our framework in a typical scenario of first responders using the application.**

## I. Introduction

Efficient communication among first responders during a disaster environment can make a difference between life and death. The effectiveness of communication among different personnel involved in incident management plays an important role in determining the outcome. An incident is managed by a dynamically formed command chain of first responders who may frequently move between different teams with different objectives. Having a well-defined naming framework, typical of the Information-Centric Networking (ICN) approach is desirable to meet these communication needs. In the past, it has been noted that publish/subscribe (pub/sub) systems (*e.g.,* [1], [2]) are convenient for information dissemination, and provide the necessary push-based information delivery needed for one-to-many first-responder communication that is typically needed in our context [3]–[5]. As a result, first responders can save significant amounts of time and mental energy by not having to identify and communicate with specific individuals and instead can communicate to the appropriate role or dynamically formed (and named) recipient sub-groups to meet their current communication needs.

Secure communication is critical for first responders to exchange vital information in an incident response. Maintaining confidentiality and message integrity are key to preventing malicious individuals from eavesdropping or impersonating. Even the general public may panic unnecessarily if information is revealed prematurely or when it is not warranted. In [6], we propose a secure communication mechanism on top of POISE [3], a pub/sub architecture that takes advantage of graph-based namespaces for efficient and flexible communication among dynamically changing first responder teams in disasters. For the implementation of the encryption mechanism, one could use either public-key encryption (PKE [7]) or attribute-based encryption (ABE) mechanisms like key-policy ABE (KP-ABE [8]) or ciphertext-policy ABE (CP-ABE [9]). KP-ABE is the best among the alternatives since it has a relatively good performance in encryption/decryption and a low maintenance overhead. Our secure communication uses a message-oriented solution with KP-ABE. The message-oriented solution leverages the namespace during *the encryption process* and embeds a recipient (group) name and its descendants into the encrypted message. By encrypting a piece of content for multiple groups, our approach enables efficient message delivery over multicast/broadcast media, unlike the use of secure unicast channels (*e.g.,* cellular environments such as FirstNet [10]) which result in the excessive and duplicate encryption and transmission of messages which introduces considerable overhead. On the receiver side, each subscriber only gets a decryption key that contains the name he/she subscribes to. It means that key delivery only happens when the subscription changes – change in the namespace does not result in key updates. This is beneficial in dynamic environments where frequent name changes occur, such as the addition or deletion of incidents or team members. The solution does not require a single key delivery in such scenarios since the change in the team membership is achieved by namespace updates instead of the subscription. Delivering private keys less frequently is also beneficial in an infrastructure-less environment, where the subscriber may not have access to the key issuer for new keys. The encryption framework also provides a flexible revocation mechanism that overcomes the challenges of key revocation in encrypted multicast communication. See [6] for further details.

## II. Demo Overview

In this demo, we implemented a communication system among first responders using a secure communication mechanism to show the feasibility and flexibility of the design. We show that even with broadcast media, the mechanism can ensure data confidentiality. We also demonstrate how it accommodates dynamic membership changes in infrastructure-less environments, flexible recipient selection, and key revocation.

Fig. 1 shows the major players and the topology of our demo. The players include: *1*) A **key issuer** that grants keys to the subscribers. It manages the subscription permissions but does not know the namespace. All communications with the key issuer are encrypted by SSH to provide authentication of the publisher/subscriber and confidentiality when delivering keys. *2*) A **hub** that simply broadcasts each message to all the receivers. The hub does not have any understanding of namespace, keys, or encryption. *3*) A single **publisher** that manages the namespace. The namespace is *only* maintained on the publisher. Neither the key issuer nor the subscribers need to know their relationships. This setting is for demo

979-8-3503-0322-3/23/$31.00 ©2023 IEEE

2023 IEEE 31st International Conference on Network Protocols (ICNP) | 979-8-3503-0322-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICNP59255.2023.10355578

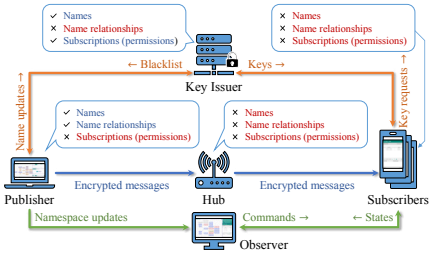
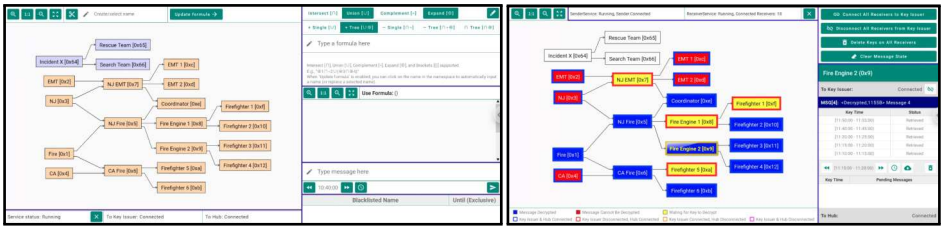



Fig. 1: Roles in Demo



(a) Publisher's View (b) Observer's View

Fig. 2: View of Different Roles

purposes only, to show the minimal required information on each role. In real-world scenarios, the system can have multiple publishers who maintain a synchronized view of the namespace. Subscribers can also be publishers. *4*) A set of **subscribers** that receive messages and try to decrypt them. The subscribers will subscribe to names and retrieve keys from the key issuer periodically. They can also be disconnected from the key issuer to show that our solution also works in disconnected environments. We also allow the subscribers to delete the keys stored locally for some scenarios. *5*) An **observer** that gets the namespace from the publisher and the decryption states from the subscribers and then aggregates them into a single view. The observer also has a GUI to help remotely control the subscribers.

Fig. 2 shows the view of the publisher and the observer. The publisher (Fig. 2a) sees the namespace on the left. He can use it to modify the group relationship and specify the recipient set. The right side includes a recipient set editor and a simple UI to send text messages. We also show the blacklisted users at the bottom right of the view just for demo purposes. The publisher does not need to know the list since the application excludes the blacklisted users automatically. The left side of the observer view (Fig. 2b) shows the namespace synchronized from the publisher. For simplicity, in this demo, we only assign one subscriber for each name. Therefore, each name in the observer view also represents the subscriber subscribing to the name. We use the border color to represent if the subscriber is connected to the key issuer: blue=connected, red=disconnected (infrastructure-less), black=no subscriber subscribing to the name. The fill color represents if the subscriber can decrypt the latest message: blue=can decrypt, red=cannot decrypt, yellow=waiting to retrieve a key (to try decrypt). On selecting a subscriber (highlighted in the namespace), the bottom right section shows the subscriber's view, including the message decryption state of the subscriber (same color profile as the fill color), the saved keys, and the pending messages (waiting for keys). We can remotely control the selected subscriber, *e.g.,* disconnect it from the key issuer, and request/delete keys. On top of the subscriber view, we listed several commands that we want to operate on all the subscribers.

## III. Demonstrated Features

In the demo, we will show the following features: *1*) **Publishing to a group utilizing graph-based namespaces:** When the publisher sends a message to a name (group), even if all the subscribers can receive the message (the hub mimics broadcast media), only the subscribers of the name and its descendants in the namespace (members or members of subgroups) can decrypt it. *2*) **Communication among dynamically changing groups in disconnected environment:** The subscribers that can decrypt the messages change automatically when the namespace is updated. Since our solution does not need to generate new keys on namespace updates, it works well even when all the subscribers are disconnected from the key issuer. *3*) **Rich recipient selection semantics:** We demonstrate how to use the recipient set editor to make use of the recipient selection semantics provided by the secure mechanism. Examples include sending a message to multiple teams, including/excluding a single user or a group, and to a set of users that share some common features (*e.g.,* firefighters that are dealing with incident X, a policeman not dealing with any incident). The publisher can create recipient selection formulas by simply clicking on the GUI or via manual input. *4*) **Key revocation:** We have implemented and demonstrated the process of key revocation using both timeout and blacklist. The system can disable a single device even when a subscriber has multiple devices subscribing to the same name. We also show that the key revocation works in disconnected environments.

## IV. Conclusion

This demo shows the feasibility and efficiency of our encryption mechanism for a name-based pub/sub communication system to achieve flexible, scalable multiparty communication maintaining confidentiality and integrity. It shows that the mechanism works well in an infrastructure-less environment with dynamically changing groups, supports flexible selection of recipients, and allows easy key revocation.

## V. Acknowledgement

This work was funded by the US Department of Commerce, NIST (70NANB17H188), and US NSF grant CNS-1818971.

## References

[1] N. Fotiou *et al.*, "Developing information networking further: From psirp to pursuit," in *BROADNETS*, 2012.
[2] J. Chen *et al.*, "Copss: An efficient content oriented publish/subscribe system," in *ANCS*, 2011.
[3] M. Jahanian *et al.*, "Graph-based namespaces and load sharing for efficient information dissemination in disasters," in *ICNP*, 2019.
[4] J. Chen *et al.*, "Cns: content-oriented notification service for managing disasters," in *ICN*, 2016.
[5] K. K. Ramakrishnan *et al.*, "Resilient communication for dynamic first responder teams in disaster management," *COMMAG*, pp. 93–99, 2022.
[6] H. Yu *et al.*, "Flexible and efficient encrypted communications for dynamic teams in disaster management," in *ICNP*, 2023, (to appear).
[7] R. L. Rivest *et al.*, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, pp. 120–126, 1978.
[8] V. Goyal *et al.*, "Attribute-based encryption for fine-grained access control of encrypted data," in *CCS*, 2006.
[9] J. Bethencourt *et al.*, "Ciphertext-policy attribute-based encryption," in *Symposium on security and privacy*, 2007.
[10] "Firstnet website," https://www.firstnet.com, [accessed on Aug.6, 2023].