A New Approach to Securing Audio Conference Tools

Zhenkai Zhu
UCLA
Los Angeles, California, USA
zhenkai@cs.ucla.edu

Paolo Gasti
UC Irvine
Irvine, California, USA
pgasti@uci.edu

Yanbin Lu
UC Irvine
Irvine, California, USA
yanbin@uci.edu

Jeffery Burke
UCLA
Los Angeles, California, USA
jburke@remap.ucla.edu

Van Jacobson
PARC
Palo Alto, California, USA
van@parc.com

Lixia Zhang
UCLA
Los Angeles, California, USA
lixia@cs.ucla.edu

ABSTRACT

Named Data Networking (NDN), a recently proposed Internet architecture based on content-centric networking, is designed to secure data directly, instead of securing the communication channel between source and destination as in today's Internet. Given the novelty of this approach, application developers face challenges in its execution: what is the best way to secure data for various different applications that are developed to run over NDN networks? In this paper we describe the design of security mechanisms for Audio Conference Tool (ACT) and show how this approach can protect against common threats. We hope this design example will help the community's understanding of security designs in content-centric networks.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Security and protection

General Terms

Security

Keywords

Named Data Networking, Data Security, Distributed Design

1. INTRODUCTION

Named Data Networking (NDN) [12] is a newly proposed Internet architecture. NDN treats data, instead of hosts, into a first-class entity, and its design aims to secure data directly, instead of securing communication channels as current protocols such as SSL/TLS [4] and IPSec [9] do. Developing an application over NDN, however, raises the question of exactly *how* to materialize this vision.

AINTEC'11, November 9–11, 2011, Bangkok, Thailand.

Our approach to addressing this challenge is to develop a series of pilot applications over NDN, each requiring a specific set of security features. One of these applications is ACT, an audio conference tool. The basic design of ACT is presented in [13], which is based on the named data paradigm to support robust audio conferences. In this paper we focus on the design of the security mechanisms of ACT.

Conventional approaches to securing audio conferences heavily rely on the centralized control. ACT moves away from centralized control to a completely distributed design, achieving source authentication, participant control, and private conferencing in the absence of a central controller.

Our design secures data communications through public key cryptography. As described in [12], NDN emphasizes the distinction between the *use* of public keys, i.e. encryption and signature verification, and *trust management*, which provides an infrastructure for trust in users' public keys. NDN assumes that each party is associated with one or multiple keys and each application uses those keys to secure data. Trust management, on the other hand, is not confined within individual applications, and is subject to different policies by different people and different organizations. Trust management can be implemented through a variety of means ranging from manually configured trust anchors, PKIs, to new approaches such as [5, 11, 10]. Therefore, trust management can and should be provided as separate and independent component.

Once the trust relationship is established, conference participants and data flows in ACT are managed through the use of public keys, rather than by setting up sessions from the central controller.

This paper is organized as follows: Section 2 provides a brief background of NDN and ACT. In Section 3, we define the security requirements for ACT while in Section 4 discuss how ACT satisfies such requirements. Our results are discussed in Section 5. We conclude in Section 6.

2. BACKGROUND

2.1 Named Data Networking (NDN)

Entities in NDN [8] identify and retrieve content using data names. Data names follow a hierarchical structure, and communication is driven by the receiving end, i.e., the data consumer. To receive data, a consumer sends out an ***Interest*** packet, which carries a name that identifies the desired data.

A router remembers the interface from which the request comes in, and then forwards the Interest packet by looking up the name in its *Forwarding Information Base (FIB)*, which is populated by routing protocols that propagate name prefixes instead of IP prefixes. If more than one Interest packets are received that carry the same data name, the router simply remembers their arrival interfaces. Once the Interest packet reaches a node with the requested data, the ***Data*** packet D is sent back. D carries the name and the data, together with a signature signed by the original data producer that binds together the name and the data. As a result of the state that was set up by the Interest packets at the intermediate routers, D traces the reverse paths back to all the data consumers that have requested the data. The router may also cache the Data packets in order to answer the later requests for the same data.

While the current Internet secures the data container (i.e., the connection between source and destination), NDN secures the content itself. This design choice decouples trust in data from trust in hosts, enabling several highly scalable communication mechanisms, such as automatic caching. Meanwhile, it also brings challenges to the current security practices.

2.2 Overview of ACT

ACT [13] is one of our efforts to explore secure application designs on NDN. Instead of relying on centralized services as current implementations do, ACT takes a named data approach to discover conferences and speakers, and to fetch voice data from individual speakers. The resulting design is completely distributed and robust against failures. ACT collects the most recent information about scheduled and ongoing conferences. This requires propagating Interest packets everywhere across the network. Hence, ACT chooses names for conference information data from a broadcast name space.

To discover a conference, ACT sends an Interest for the name `/ndn/broadcast/conference/conference-list`. Any user announcing a conference should reply to this Interest with a Data packet describing the conference information in the *Session Description Protocol (SDP)* [6] format, including estimated starting time, media type supported, etc. The name of the data is constructed by appending a unique conference name component to the name carried in the Interest packet. Each ACT user always keeps an outstanding Interest for conference discovery, so that new or updated conference data can be fetched as soon as they are generated.

For each ongoing conference that a user wants to join, the next step is to collect the information of all active speakers so that the user can send Interest packets to retrieve their voice data. Speaker discovery of a particular conference is done in a way similar to conference discovery, i.e. each user sends a broadcast Interest that can reach all the active speakers; each of them then sends a speaker description data packet in reply. Included in the speaker description data are the site-dependent name prefixes used for voice data, the codecs and rates of the audio streams, among other information. Once a user acquires the information related to a conference, he/she can receive the voice data by sending Interests directly towards each speaker.

3. ACT SECURITY REQUIREMENTS

Due to the large number of scenarios in which ACT can be used, different conferences may have different security requirements. ACT is designed to provide the following security guarantees:

- *Data Authenticity:* ACT must provide data authenticity. Conference participants must be able to verify that each piece of audio data is generated by the intended source, as indicated on the data packet. This level of security is required in any kind of conference, from public meetings open to anyone (e.g. IETF meetings) to private conference calls.
- *Participant Control:* In addition, some conferences require the ability to control the list of participants. We indicate users who are not part of the conference as "outsiders". Outsiders must not be able to listen or to inject voice data into ongoing conferences.
- *Anonymity:* Private conferences require the ability to hide the list of participants to outsiders, who must not be able to learn whether a user is a member of a conference.

In the next section, we show how ACT fulfills these requirements.

4. SECURING ACT

In ACT, the user who creates a conference is called "Organizer". Organizer is the only entity that has permission to change the conference description, to add or remove participants and to devise and enforce the participant control policies. ACT security design is based on the following assumptions:

- A trust management system, which allows applications to determine the validity of the public keys, is provided by the underlying NDN layer or by some other mechanism (see e.g. [5, 11, 10]).
- Organizer knows the identity of all users who are allowed to join the conference it creates.
- Participants are semi-trusted, i.e., they follow the protocol faithfully but may try to learn additional information from their interaction with other users. We make no assumptions on the behavior of a former participant once he leaves a conference.

The rest of this section illustrates the supporting mechanisms to meet the ACT security requirements as described in Section 3.

4.1 Data Authenticity

ACT security relies on NDN basic functionalities for data authentication and for encryption based access control. NDN data packets, and therefore also ACT data packets, are digitally signed, and the name in each NDN data packet is cryptographically bound to the corresponding packet content. When participants are not required to be anonymous, this ensures both the integrity and authenticity of each packet. In case of anonymous participants, data authenticity issues are discussed in Section 4.4.

4.2 Participant Control

ACT employs an encryption-based access control scheme that allows only the eligible participants to decrypt the information about the conferences.

For a conference that requires participant control, its Organizer generates a public/private key pair (K_e, K_d), where K_e is used for encryption while K_d is used for decryption,

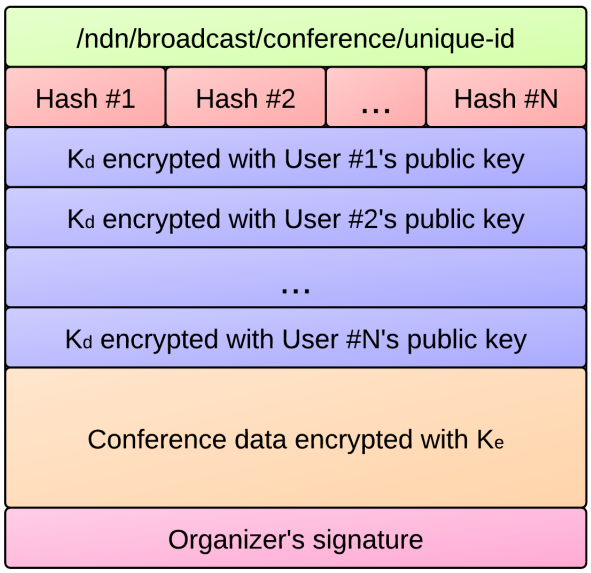

Figure 1: Participation-control Enabled Conference Announcement Data

to distribute confidential information within the conference. Organizer keeps the encryption key secret and distributes the decryption key K_d to all legitimate participants. Conference information is encrypted by Organizer using K_e and can only be accessed by those who obtain K_d.

K_d is sent to conference participants in encrypted form in order to prevent outsiders from accessing it. The hash values of the eligible participants' public keys are also included together with the encrypted K_d. In this way, each user can determine whether he/she is among the legitimate participants without performing any decryption.

Only Organizer, who knows K_e, can further update the conference information or alter participant control policies. In order to fulfill this requirement, the underlying encryption scheme must prevent users with the knowledge of K_d to determine the value of K_e. This can easily be achieved using RSA-OAEP [3]. In particular, given $N = p \cdot q$ where p and q are safe primes, in our instantiation the encryption exponent e (only known to Organizer) is chosen uniformly at random from all the values $1 < e < \phi(N)$ such that e is co-prime with respect to $\phi(N)$. Unfortunately, this does not allow us to adopt some of the common optimizations related to RSA. It is not possible to select an exponent e with low hamming weight, since participants would be able to determine its value based on the knowledge of N. Moreover, since the knowledge of p and q allows any party to compute the e from the decryption exponent d, only Organizer can use CRT to speed up RSA operations.

An example of a typical participation-control enabled conference announcement data packet is shown in Figure 1. All encryptions of K_d (one per participant) are included in a single data packet. While this slightly increases the cost of retrieving K_d for participants, it allows a better utilization of the natural multicast and caching capabilities of NDN.

4.3 Voice Data Encryption

Voice data is generated with significantly higher frequency and in considerably higher volume compared to conference announcement data. Moreover, while there is only one Organizer for each conference, there may be multiple speakers. In fact, it is not uncommon that all participants in a small conference speak at some time. Hence, the approach used in securing conference data may not scale well enough to cope with voice data. There are mainly two reasons for not using asymmetric encryption for voice data: 1. According to the protocol above, each speaker has to generate a key pair (K_e^s, K_d^s) and distribute K_d^s to all listeners. This requires each speaker to have complete knowledge of the other participants in the conference as well as their public keys, which is often not the case. Besides, distributing a private key for each speaker also incurs a significant amount of overhead; 2. Asymmetric encryption imposes a significantly higher computation overhead compared to symmetric encryption. Doing asymmetric encryption for each voice data packet raises concerns about the computation overhead, especially on devices with limited resources (e.g. smart phones, tablets).

For this reason, ACT relies only on symmetric keys for voice data encryption. Organizer establishes the key for voice data, and participants use the same key to decrypt data from speakers and encrypt their own packets.

4.4 Participant Identity Protection

In some cases, participants may wish to keep their identity hidden from outsiders. In particular, outsiders should not be able to tell whether a user is participating in a given conference. Hence, the hash values of the participants' public keys, which are used to help identify legitimate participants, should no longer be included in the conference announcement data packet. Moreover, K_d must be encrypted using a key-private encryption scheme [2], so that observers cannot determine the identity of participants' public key by observing an encrypted conference announcement.

Speakers should also generate a temporary asymmetric key pair for signing speaker information data, so that the signatures in the NDN packets will not reveal their identities. Speakers must include signatures that guarantee the authenticity of their voice data encrypted together with the speaker information data using the symmetric keys of the conference. Figure 2 shows the structure of a voice data packet from a speaker.

Although a site-dependent prefix inevitably reveals the topological location of a user (which may be related to participants' physical location), a third party cannot distinguish voice data packets from other packets due to encryption. Therefore, no external adversary can tell which names are used in ACT for audio streams. In order to achieve better anonymity, users should rely on NDN anonymizing techniques to access and publish conference data.

4.5 Key Revocation

Organizer uses key revocation to force one or more participants to leave the conference. In ACT, key revocation is straightforward. As ACT keeps an outstanding Interest for new or updated conference description [13], Organizer can generate an announcement at any time for the key revocation. All the participants that are still eligible for the conference will fetch the updated keys immediately.

In order to distribute a new asymmetric key pair for a conference, Organizer uses the current K_e to encrypt the conference announcement, which indicates the asymmetric key revocation and includes normal conference information

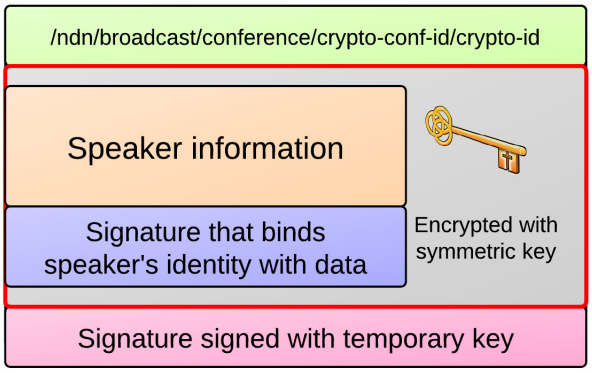

Figure 2: Speaker Information Data Packet

encrypted with an new key K'_e. K_e is used to encrypt the data so that the participants are assured that the key revocation is legitimate, as the conference Organizer is the only one who knows K_e. The recipients then check whether they are still allowed to participate and, if they are, successfully decrypt K'_d.

To issue a new symmetric key, which supersedes the current one, Organizer simply includes the new key in the conference announcement.

5. DISCUSSIONS

In this section we discuss some of our choices in the ACT security design.

5.1 Secret Participants List

As mentioned in the previous sections, there are some circumstances in which concealing the identity of participants is desirable. In our current design, this simple difference in requirements leads to significant changes in the processing overhead of conference participant discovery. Because no information about the participants can be disclosed in public, the design described in Section 4.4 forces all the recipients to go through a trial-and-error process of decrypting each encrypted K_d in order to determine whether they are eligible to join the conference. This process can lead to serious scalability concerns as all the users within the conference broadcast scope have to spend their computational power to determine their eligibility for a conference, and a large conference would have a long list of encrypted K_d list. A possible solution to this issue is the use of broadcast encryption [7] for large conferences with secret participants lists. This allows participants to determine whether they are allowed to join the conference by performing one single decryption. However, we consider the cost related to broadcast too high compared to the approach described above for conferences with less than a few hundreds participants.

5.2 Use of Symmetric Keys

The use of symmetric key encryption instead of public key encryption eliminates the need for every speaker to distribute a decryption key to all conference participants, which introduces a non-negligible overhead. It also alleviates the computational cost of encrypting and decrypting data. On the other hand, symmetric keys have a critical limitation: due to their symmetric nature, they do not enforce any distinction between data producers and consumers. Any user, given the right to decrypt, can also encrypt data using the same symmetric key. In multi-party communications, symmetric keys should be used with caution as an engineering optimization, rather than the primary tool. We use symmetric keys for voice data encryption to reduce the computational burden of ACT based on the assumption that every participant has the permission to speak and participants are semi-honest, i.e., they will not impersonate each other. The fact that it is usually possible to distinguish or recognize people by their voice also adds another reason for choosing symmetric keys for such purpose.

6. CONCLUSION

In this paper we presented the design of the security mechanisms for ACT, a distributed audio conference tool over NDN. This design uses only simple cryptographic tools, but represents a fundamental departure from conventional approaches which rely on centralized controllers and session-based security. Through directly securing data rather than its containers as well as a strong separation between encryption/authentication and trust management, our design meets the security requirements in a distributed way and enables each conference group to devise and enforce their own security policies. The design as reported in this paper has been implemented in ACT. Interested readers can find the implementation at [1].

We hope the work reported in this paper can serve as an illustrative example to show how one may benefit from NDN's basic machinery of securing data directly to develop secure applications in a simple and straight forward way. We are currently working on a trust management system design which is expected to be implemented in near future.

7. REFERENCES

[1] `https://zhenkai@github.com/zhenkai/mumble.git`.
[2] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT*, 2001.
[3] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *EUROCRYPT*, 1994.
[4] T. Dierks and C. Allen. The TLS protocol version 1.0. RFC2246. 1999.
[5] P. Gutmann. Plug-and-play pki: A pki your mother can use. *Proceedings of the 12th USENIX Security Symposium*, August 2003.
[6] M. Handley, V. Jacobson, and C. Perkins. Sdp: Session decription protocol. *IETF RFC 4566*, July 2006.
[7] J. Horwitz. A survey of broadcast encryption. Technical report, 2003.
[8] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard. Networking named content. *ACM CoNext'09*, December 2009.
[9] S. Kent and K. Seo. Security architecture for the internet protocol. *RFC 4301*, December 2005.
[10] E. Osterweil, D. Massey, B. Tsendjav, B. Zhang, and L. Zhang. Security through publicity. *HOTSEC'06*, 2006.
[11] D. Wendlandt, D. Anderson, and A. Perrig. Perspective: Improving ssh-style host authentication with multi-path probing. *Proc. USENIX Annual Technical Conference*, June 2008.
[12] L. Zhang et al. Named data networking (NDN) project. *Technical Report NDN-0001*, October 2010.
[13] Z. Zhu, S. Wang, X. Yang, V. Jacobson, and L. Zhang. Act: An audio conference tool over named data networking. *ACM ICN'11*, August 2011.