Moderator-controlled Information Sharing by Identity-based Aggregate Signatures for Information Centric Networking

Tohru Asami
The University of Tokyo
Tokyo 113-8656, Japan
asami@akg.t.u-tokyo.ac.jp

Byambajav Namsraijav, Yoshihiko Kawahara
The University of Tokyo
Tokyo 113-8656, Japan
{byambajav, kawahara}@akg.t.u-tokyo.ac.jp

Kohei Sugiyama, Atsushi Tagami
KDDI R&D Laboratories
Saitama, 356-8502 Japan
{ko-sugiyama, tagami}@kddilabs.jp

Tomohiko Yagyu
NEC Corporation
Kawasaki, 221-8666 Japan
yagyu@cp.jp.nec.com

Kenichi Nakamura
Panasonic Corporation
Tokyo 105-8301, Japan
nakamura.kenken@jp.panasonic.com

Toru Hasegawa
Osaka University
Osaka, 565-0871, Japan
t-hasegawa@ist.osaka-u.ac.jp

ABSTRACT

Information sharing services have been provided via common servers, which not only relay messages but also sometimes moderate them. A peer can become a moderator and control the distribution of messages belonging to his private message group. However, the physical transfer of a message is usually out of the peer's control. Originator-signed signatures inherent in Information Centric Networking assure the integrity and provenance of messages exchanged among peers, which makes it possible to realize moderator-controlled information sharing in which a peer can become a moderator and control the distribution of his private message group as a trustable server. However, moderated content requires multiple signatures, which increases the size of the exchanged message and is inadequate, especially for short message services. We propose the use of Identity-Based Aggregate Signatures (IBAS) to decrease this overhead, and provide a proof-of-concept IBAS implementation for Named Data Networking (NDN). We also compare the performance of the proposed IBAS implementation with existing RSA signatures. An overhead reduction of approximately 45% to 60% compared to RSA signatures is achieved for an NDN packet in the proposed configuration. Because of the properties of the identity-based signature[24], this IBAS implementation is robust and works even during a disaster or when a trustable centralized server is not online.

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.

ICN'15, September 30–October 2, 2015, San Francisco, CA, USA.
© 2015 ACM. ISBN 978-1-4503-3855-4/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2810156.2810163.

Categories and Subject Descriptors

C.2.1 [**Computer-communication networks**]: Network architecture and design-distributed networks

General Terms

Performance, Design

Keywords

named data networking; identity-based aggregate signatures; moderator-controlled information sharing

1. INTRODUCTION

Information sharing through moderators has been common since the 1980's. NetNews [15] and mailing lists such as LISTSERV [16] and Majordomo [2] are typical examples. They, as well as many modern social networking services (SNSs) such as Facebook [11], have a functional division of their operations between administrators and moderators. The tasks of the administrator in these services are to change list or newsgroup settings and to add moderators and subscribers, whereas the tasks of the administrator in modern social networking services (SNSs) such as Facebook [11] are to, for example, adding/deleting peers, content storage of uploaded messages, and time-stamping received messages. The moderator's role in NetNews is to supervise posted messages to the list or the newsgroup, and the posted message will not be sent to the subscribers unless approved by the moderator, whereas each SNS peer in the SNS controls the circulation of the moderator's messages by creating friend lists or groups, which are lists of friends gathered based on intention. Once created, each circulation is usually performed without any human interventions. A server managed by an administrator is considered to be a trustable message exchange point among related peers or a kind of portal site for this SNS that is known to everybody in the serving network. In other words, the SNS administrator can observe any message sent to the public or to a specific friend group. From a privacy point of view, a future SNS service should give a peer the discretion to decide whether his messages are physically transferred only within his own closed-user group without the help of the SNS administrator's server.

Originator-signed messaging introduced by Information-Centric Networking (ICN) [28] makes it possible for each peer to become not only a moderator but also a trustable server for his friend lists by assuring the integrity and provenance of the received content. We refer to this as moderator-controlled information sharing (MIS). An MIS service is served by several moderators, and this decentralized MIS is robust in a disaster scenario, allowing safety confirmations to be exchanged among family members [26]. Such confirmations will be successful if (1) the moderator for this family exists on a network in the disaster area and (2) authentication can be performed without a trustable server online. Condition (1) will be frequently satisfied because such a friendship relation is often geographically constrained. Thus, if condition (2) is satisfied, ICN-based MIS can not only solve the privacy concerns of the current SNS but can also increase the robustness of the service as a lifeline infrastructure.

However, at least two signatures are required to convince the subscriber of the authenticities of the content publisher and the moderator. Attaching multiple signatures to a single short message increases the message size and may not be suitable for human-generated content exchanges, such as Facebook, from the point of view that the specification of short message service (SMS) by GSM/W-CDMA requires the number of message characters to be at most 140 [1]. Thus, in addition to condition (2), a method of minimizing the *Data* packet size in the above-mentioned moderator-controlled short message sharing is needed. In order to solve the above-mentioned problems, we propose the use of Identity-Based Aggregate Signatures (IBAS) [13] as yet another signature scheme for Named Data Networking (NDN), while maintaining its compatibility with the normal version of NDN. We implement this signature scheme on an existing open-source NDN implementation ndn-cxx [18] and compare its performance with the traditional Public Key Infrastructure RSA signatures. Although there have been proposals for using IBS/IBE in NDN [29], [22], to the best of our knowledge, our proposal is the first implementation of IBAS in any kind of network.

The remainder of the present paper is organized as follows: Section 2 introduces NDN, Identity-Based Aggregate Signatures, and research related to NDN security. Section 3 presents the definition of ICN-based MIS and discusses inherent problems, followed by the proposed solution in Section 4, the implementation design for NDN in Section 5, experiments in Section 6, and a discussions in Section 7. Finally, we conclude the paper in Section 8.

2. RELATED RESEARCH

This section presents a brief overview of the technologies related to our ICN-based MIS.

2.1 Named Data Networking (NDN)

Overview

Named Data Networking (NDN) [28] is a type of ICN in which communication is based on named contents, rather than host addresses, and every named content is identified, addressed, and retrieved by its name instead of its physical location (i.e., host address). In the conventional Internet, security is achieved in end-to-end connections. On the other hand, NDN routers cache contents and serve contents directly from themselves in order to achieve efficient content delivery. Therefore, in NDN, security is built into contents, rather than connections between end hosts. Consequently, content publishers must sign all of their contents, i.e., any content in NDN has a signature to prove its provenance and integrity.

Packet Structure

As specified in [20], NDN's packet encoding employs the Type-Length-Value (TLV) format [21]. An NDN packet is primarily a collection of TLVs inside a top TLV element (either *Interest* or *Data* packet), and each sub-TLV can also be further nested. *Interest* consists of five TLV elements, i.e., *Name*, *Selectors*, *Nonce*, *Scope*, and *InterestLifetime*, whereas *Data* consists of four TLV elements, i.e., *Name*, *MetaInfo*, *Content*, and *Signature*. The NDN packet does not have a fixed packet header.

Forwarding

In order to obtain content, a consumer placed the name of the desired data into an *Interest* packet and sends it out. Routers use this name to forward the *Interest* toward the intended data producer(s). The forwarding decisions at routers in this step are based on their Forwarding Information Base (FIB). While forwarding the *Interest* packet, the routers register it in their Pending Interest Table (PIT). Once the interest reaches a node that has the requested data in its Content Store (CS), the node will return a *Data* packet that has the same name as the received *Interest* and contains the requested content. This *Data* packet also contains a signature that proves that the data was indeed produced by the intended producer and has the name requested by the *Interest*. The returned *Data* packet follows in reverse the path taken by the *Interest*. On the reverse path, routers refer to their PIT to forward the returning *Data* packet to the requesting consumer. Also, the routers cache the *Data* into their CS so that they can serve it directly upon further requests.

2.2 Existing Security Proposals in NDN

In this subsection, we introduce the current key management models. Current NDN testbed key management [3] uses a simple hierarchical trust model with a root key, which signs the keys for each site. Then, the sites' keys are used to sign their users' keys, and the users' keys are used to sign their devices (Fig. 1). This trust model is essentially the same as the traditional Public Key Infrastructure (PKI) model, in which the testbed root is equivalent to a widely trusted global certification authority (CA), and the sites are equivalent to middle layer CAs. In order to verify a *Data* packet's signature signed by the ordinary RSA or Elliptic Curve Digital Signature Algorithm (ECDSA), the verifier must trace all of the parents' public keys and certificates until reaching the NDN testbed root.

There is an approach to using a Web-of-Trust in conjunction with self-certifying names for a fragmented (mobile) networks scenario [23], where connectivity to centralized en-

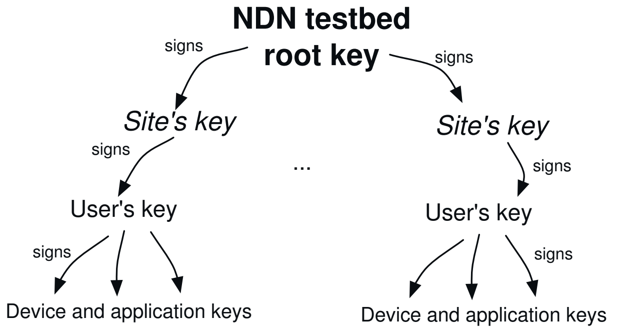

Figure 1: Key trust model on the NDN testbed [3]

tities and authentication servers are not available. However, the signature scheme itself is as described above.

2.3 Identity-based Aggregate Signatures (IBAS)

Boneh et al. proposed the concept of aggregate signatures in [4], in which it is possible to aggregate n signatures on n distinct messages from n distinct users into a single aggregated signature of constant size. This single signature (and the n original messages) will convince the verifier that the n users did indeed sign the n original messages (i.e., user i signed message m_i for $i = 1, ..., n$). The main motivation for using aggregate signatures is compactness. However, in the case of a non-identity-based aggregate signature scheme, such as that proposed in [4], verification of an aggregated signature requires the verifier to obtain the public keys of all participating signers. Furthermore, in order to verify the public keys, the verifier may need to obtain their parent certificates. Since the sizes of public keys are as large as those of signatures, the necessity for obtaining public keys and parent certificates largely negates the advantage of aggregate signatures.

Gentry et al. proposed an Identity-Based Aggregate Signatures (IBAS) scheme [13] to solve the above shortcoming of aggregate signatures. The IBAS scheme combines the Identity-Based Signature (IBS) and the aggregate signature. In the IBAS scheme, the verification information (apart from the description of who signed which message, i.e., the list of identities and their messages) consists only of a single aggregate signature and the public parameters given by the Private Key Generator (PKG). The following describes in detail the construction of the IBAS scheme (using the same notation as [13]). Compared to a normal non-aggregate IBS scheme, such as CC-IBS [7], the IBAS scheme has an extra **Aggregation** step.

Setup: The Private Key Generator (PKG)

(a) generates (elliptic curve) groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q and an admissible bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$;
(b) chooses an arbitrary generator element $P \in \mathbb{G}_1$;
(c) chooses a random $s \in \mathbb{Z}/q\mathbb{Z}$ and calculates $Q = sP$;
(d) chooses three cryptographic hash functions: $H_1, H_2 : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}/q\mathbb{Z}$.

As a result, the system parameters are $params = (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, Q, H_1, H_2, H_3)$, and the PKG's secret key is $s \in \mathbb{Z}/q\mathbb{Z}$.

Private Key Extraction: By authenticating itself to the PKG, the client with identity ID_i receives the values of $P_{i,j} = H_1(\mathsf{ID}_i, j)$ for $j \in \{0, 1\}$.

Individual Signing: The first signer chooses a unique string (state information) w that has never been used before. In order to sign a message m_i, the signer with identity ID_i:

(a) computes $P_w = H_2(w) \in \mathbb{G}_1$;
(b) computes $c_i = H_3(m_i, \mathsf{ID}_i, w) \in \mathbb{Z}/q\mathbb{Z}$;
(c) generates a random element $r_i \in \mathbb{Z}/q\mathbb{Z}$;
(d) calculates the signature tuple (w, S'_i, T'_i), where $S'_i = r_i P_w + sP_{i,0} + c_i s P_{i,1}$ and $T'_i = r_i P$.

Aggregation: Individual signatures $(w, S'_i, T'_i)_{1 \leq i \leq n}$ are aggregated into an aggregate signature $(w, \quad S_n = \sum_{i=1}^{n} S'_i, \quad T_n = \sum_{i=1}^{n} T'_i)$.

Verification: The verifier checks the following equality.

$$\hat{e}(S_n, P) = \hat{e}(T_n, P_w)\hat{e}(Q, \sum_{i=1}^{n} P_{i,0} + \sum_{i=1}^{n} c_i P_{i,1}), \qquad (1)$$

3. ICN-BASED MODERATOR-CONTROLLED INFORMATION SHARING SERVICE

In this section, we present an overview of our target application, moderator-controlled information sharing (MIS), and its implementation by ICN, and then discuss two problems involved in building such an application.

3.1 Overview of MIS

Our simplified model for an MIS service consists of at least three different types of participants: moderators, peers, and an administrator. Peers are content/message publishers as well as subscribers within their preferred interest groups or subscription lists. A peer can become a moderator for a specific group and can supervise posted messages to this group. In Fig. 2, the group supervised by $Moderator_1$ is indicated by solid arrows to its subscribers. The moderator can control the information released to this group. The posted message will not be sent to the subscribers unless approved by the moderator. A moderator can join a group created by another moderator as a subscriber. In Fig. 2, $Moderator_2$ has joined the group of $Moderator_1$. Another role of the moderator is to relay posted messages to related peers or subscribers if desired. In this case, moderators relay messages, like traditional SNS servers managed by administrators. Of course, a moderator can delegate this function to the administrator if desired. Thus, the administrator is a moderator who supervises a special 'public' group, indicated by the dashed arrows from the moderator to all of the peers. This group is a kind of portal site for the MIS service, and so MIS can be defined as a service that has evolved from the current centralized information sharing services.

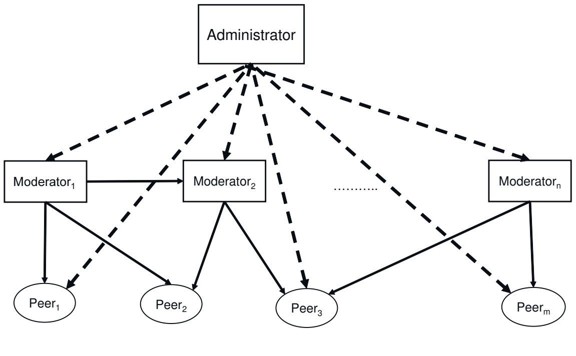

Figure 2: Structure of the MIS service

Messages in the MIS service can be relayed by a set of moderators including an administrator. This distributed approach has two advantages that current information sharing services in the Internet lack.

1. **Privacy:** Each moderator has the discretion to decide whether messages are physically transferred to the administrator or relayed by himself. In the latter case, the moderator can physically relay messages independent from the administrator.
2. **Fault tolerance:** All peers can communicate with each other within their preferred group when their relaying moderator is reachable by them. Even if the connection to the administrator is lost, it only prevents information sharing through the 'public' group. In such a case, Fig. 2 has three information exchange groups: $Group_1 = \{Moderator_1, Peer_1, Peer_2, Moderator_2\}$, $Group_2 = \{Moderator_2, Peer_2, Peer_3\}$ and $Group_3 = \{Moderator_n, Peer_3, Peer_m\}$. Thus, communications are possible via nodes in the intersections of these groups. For example, $Peer_m$ can communicate even with $Peer_1$ if human interventions at $Peer_3$ and $Peer_2$ exist. This advantage is very useful for safety confirmation exchanges during a disaster when the entire network is not well connected.

The requirements to achieve MIS are as follows.

1. Moderators must be trusted by their peers in any situation.
2. A peer can join a group managed by a moderator or can even become a moderator at his convenience.

These requirements must be satisfied even in a fragmented network just after a disaster.

3.2 ICN-based MIS

Information-centric networking is a promising technology for realizing the above mentioned MIS, and we refer to the result as ICN-based MIS. First, originator-signed signatures in ICN assure integrity and provenance for messages exchanged by peers. Authentications and authorizations are very easy when peers join moderators' groups. Second, any peer can be a moderator if the binding among his identifier (ID), public key, and secret key is trustable. Third, the existence of ICN network caches can reduce the load of the peer when the peer becomes a moderator. Finally, name-based routing inherent in ICN makes it possible to deliver messages from peers to their moderator even if global connections in datalink and network layers are lost. In this case, although routing is a concern for ICN, it is beyond the scope of the present paper. We herein focus on ICN-based MIS from the standpoint of message provenance, assuming routing issues are not a concern.

Information-centric networking trustability is currently supported by a real-world agent according to the traditional Public Key Infrastructure (PKI) model, consisting the hierarchy from the widely trusted global certification authority (CA) to the edge CAs. If a trustable online server is required to verify the message, fault tolerance of the service cannot be achieved. In the traditional PKI, the verification needs to obtain the public keys of both the publisher and the moderator, as well as all of the parent certificates. Fetching the required public keys and certificates results in a long delay, especially in high-latency networks, such as a fragmented network in a disaster area [26].

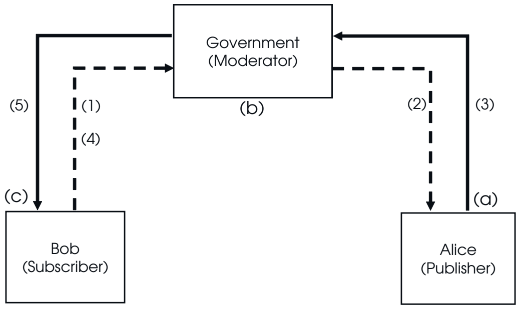

Figure 3: Information exchanges in the MIS service

Fig. 3 shows a scenario for using MIS for safety confirmation exchanges, where one peer (Alice) publishes a message and the other (Bob) receives the message or subscribes to a message service via their moderator (Government). The exchange sequence is as follows. (1) The subscriber (Bob) declares his interest through his subscription request. (2) A moderator (Government) polls the publisher. (3) If the publisher (Alice) has content to publish, she sends it to the moderator. Upon receiving the new content, the moderator first checks the content, then adds some information, such as the accepted date and the publisher's rating. (4) The subscriber requests to receive the message. (5) The moderator sends the content to corresponding subscribers or peers within his group.

In order to assure that the message is relayed via this moderator, the subscriber should have the ability to verify the authenticity of both the moderator and the publisher from the received content. Sharing a symmetric subscription key among the moderator and its peers requires additional statefull procedures among the moderator and peers, and is not suitable for use in disaster situations because connections among the moderator and peers may not be stable and it is not possible to know who will join which group beforehand. Applying a group membership to this problem has the same disadvantages.

In order to achieve this requirement using the ordinary RSA or Elliptic Curve Digital Signature Algorithm (ECDSA) signature, the moderator must attach two signatures: the signature generated by the publisher on the original content and the signature generated on the new moderated content. In the safety confirmation system [26], the content size is usually small and does not exceed 100 bytes. Most human-generated content such as that on Facebook consists of short messages, and such applications fall into a group called instant messengers. Short message transfers are also important in many sensor networks such as smart metering networks, in which the size of a message is usually small. Apart from a signature value block, the `Signature` element of a *Data* packet also contains a signature info block (Section 5.3), which contains a description of the signature, the signature algorithm used, and any other relevant information in order to obtain parent certificate(s) (e.g., `KeyLocator` ([20])). Therefore, the overhead of transmitting two signatures to deliver a single short message is huge compared to the message's size. Furthermore, if further information retrievals such as parent certificate(s) are required, received packets cannot be verified until all of the required information is received. When trustable servers are accessed in disaster situations, the connections to those servers may not be guaranteed.

The above considerations indicate that there are four requirements for message provenance in ICN-based MIS.

1. For the instantaneous reception of a message, the fetching overhead for public keys and their certificates must be reduced.
2. Verification must be achieved without any trustable servers online.
3. It is preferable to achieve a group membership using only given public key/secret key pairs.
4. It is important to reduce the signature overhead for short messages.

4. PROPOSED ICN-BASED MIS

The first and second requirements, as described for the public key retrieval delay problem in Section 3.2, can be solved using an identity-based signature scheme, such as CC-IBS [7]. As shown in Fig. 4, the proposed ICN-based MIS system adds a fourth entity, namely the Private Key Generator (PKG), to Fig. 3, which gives three other entities their secret keys corresponding to the registered identifiers (IDs) as well as the public parameters that are known to all in the network. The binding between the ID (ID_1) and the corresponding secret key ($SecretKey_1$) is assured by a physical negotiation by the signer ($Signer1$) and the PKG in our real-world activities. Once the message is known to be signed by the private key corresponding to the identifier in a received message, there is no need for the PKG to be online and instantaneous verification is achieved.

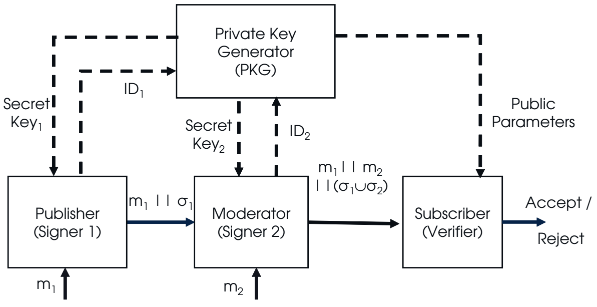

Figure 4: IBAS signing process for a message relayed by the moderator

The third and fourth requirements in Section 3.2 can be satisfied using IBAS, which is an identity-based signature. Either m_1 or m_2 in Fig. 4 indicate the signed part in the packet sent from a publisher or a moderator, which includes Name, MetaInfo, Content, and SignatureInfo. Here, σ_1 is a signature created from m_1 and $SecretKey_1$. Moreover, $\sigma_1 \cup \sigma_2$ is the aggregated signature created by σ_1, m_2, and $SecretKey_2$. In order to verify the message, a concatenation $m_1||m_2||\sigma_1 \cup \sigma_2$ must be sent to the subscriber. If m_1 can be reconstructed from m_2 in a formulated manner, sending $m_2||\sigma_1 \cup \sigma_2$ is sufficient for the verification process.

In IBS, the only information required for a verifier to check data integrity is the identity of the data producer, i.e., the publisher or the moderator, which in most cases is already known to the verifier. In NDN, in particular, the content producer's identity can be obtained from the content name. Therefore, we used IBAS, the combination of IBS and aggregate signatures, to build our MIS. In the next section, we present one implementation example for the proposed concept using NDN.

5. IMPLEMENTATION OF IBAS FOR NDN

In order to verify the *Data* packet, two different message parts signed by the publisher and the moderator are derived from the original packet. This section presents a simple example for such an implementation.

In order to measure the round trip time (RTT) in Section 6.2, the scenario of Fig. 3 is modified slightly as follows. Two peers, Bob and Alice, join the list owned by a moderator 'Government'. Through this moderator, Alice informs Bob that her new message is ready to send if requested. Bob wants to obtain a safety confirmation from Alice, and he sends an interest packet to the moderator (1). Since the safety confirmation from Alice is not yet uploaded, 'Government' sends an interest packet to Alice (2). Then, Alice replies to 'Government' by the *Data* packet with her information (3). After adding a time stamp and the moderator's name to the content part of the *Data* packet, this moderator returns the *Data* packet with the safety confirmation from Alice to Bob (4). We refer to the above scenario as the packet relay model.

Table 1 summarizes the processing performed by RSA and IBAS for the packet at each step in Fig. 3.

Table 1: Details of processing by (a) Publisher, (b) Moderator, and (c) Subscriber

	RSA	IBAS
(a)	data creation & signature generation	data creation & signature generation
(b)	signature verification & signature creation & signature concatenation	signature verification & signature generation & signature aggregation
(c)	signature verification x 2	aggregated signature verification

5.1 Naming

In NDN, naming is application-dependent, i.e., applications can design the naming system to be useful for their intended purpose. Snippet 1 contains examples of the names used in our application. Numbers (1) to (4) in Snippet 1 correspond to the numbers(1) to (3) and (5) shown in Fig. 3. (1) and (2) are the *Interest* packets sent from Bob to the moderator, 'Government', and from the moderator to Alice, and (3) and (4) are the corresponding *Data* packets generated by Alice and the moderator. In this case, '2' in (3) and '9' in (4) are the sequence numbers of the message at Alice and at the moderator, respectively. In our application, the two foremost name components of `Name` describe the content generator's identity, e.g., `/moderators/Government` and

`/wonderland/Alice`. The third name component is always the application name: `/safetyConfirmation`.

Snippet 1 Names of the packets shown in Fig. 3

```
(1) /moderators/Government/safetyConfirmation/
                wonderland/Alice
(2) /wonderland/Alice/safetyConfirmation
(3) /wonderland/Alice/safetyConfirmation/2
(4) /moderators/Government/safetyConfirmation/
                wonderland/Alice/2/9
```

Using the above naming rule, it is possible to obtain the original publisher's name from data received from a moderator, i.e., from name (4) we can derive name (3).

5.2 Message Reconstruction

Fig. 5 shows an example of the current IBAS signature for an NDN *Data* packet by the moderator, which is compared with the corresponding RSA signature. After modifying the content name, the moderator adds the dashed parts to m_1 to make m_2. In other words, two lines starting from 'From' and 'Published' are added by Alice, and two more starting from 'Moderator' and 'Accepted' are added by the moderator. In the case of RSA, the signature by the publisher is moved into the content part of the *Data* packet sent from the moderator. This is a kind of layer violation, while the IBAS *Data* packet has only one aggregated signature. Fig. 5 also shows the size of each part in an NDN *Data* packet for RSA-2048 and IBAS as discussed in Section 6.1.

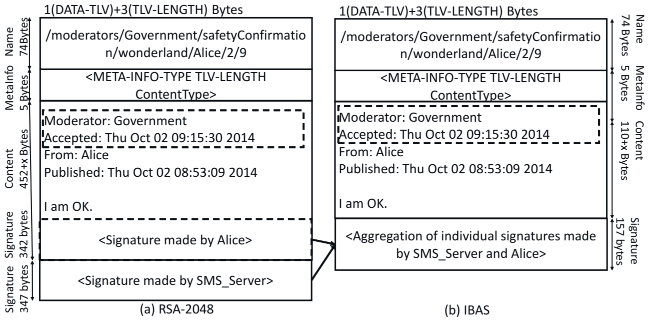
1(DATA-TLV)+3(TLV-LENGTH) Bytes
Name 74Bytes
/moderators/Government/safetyConfirmation/wonderland/Alice/2/9
MetaInfo 5 Bytes
<META-INFO-TYPE TLV-LENGTH ContentType>
Content 452+x Bytes
Moderator: Government
Accepted: Thu Oct 02 09:15:30 2014
From: Alice
Published: Thu Oct 02 08:53:09 2014
I am OK.
Signature 342 bytes
<Signature made by Alice>
Signature 347 bytes
<Signature made by SMS_Server>
(a) RSA-2048
1(DATA-TLV)+3(TLV-LENGTH) Bytes
Name 74 Bytes
/moderators/Government/safetyConfirmation/wonderland/Alice/2/9
MetaInfo 5 Bytes
<META-INFO-TYPE TLV-LENGTH ContentType>
Content 110+x Bytes
Moderator: Government
Accepted: Thu Oct 02 09:15:30 2014
From: Alice
Published: Thu Oct 02 08:53:09 2014
I am OK.
Signature 157 bytes
<Aggregation of individual signatures made by SMS_Server and Alice>
(b) IBAS

Figure 5: Structure of a *Data* packet sent from a moderator to a subscriber

5.3 Signature Structure for IBAS

The Signature element inside an NDN *Data* packet consists of two consecutive blocks as described below[20].

Snippet 2 Signature of NDN *Data* packet [20]

```
Signature ::= SignatureInfo
              SignatureBits
SignatureInfo ::= SIGNATURE-INFO-TYPE TLV-LENGTH
                    SignatureType
                    ... (SignatureType-specific TLVs)
SignatureValue ::= SIGNATURE-VALUE-TYPE TLV-LENGTH
               ... (SignatureType-specific TLVs and BYTE+)
SignatureType ::= SIGNATURE-TYPE-TYPE TLV-LENGTH
                    nonNegativeInteger
```

The size of IBAS's `SignatureInfo` is fixed (5 bytes) just to indicate the `SignatureType` in a non-negative integer, whereas that of RSA's is variable to tell the `KeyLocator` information. The `nonNegativeInteger` value in `SignatureType` represents the type of signature algorithm used in signing, as shown in Table 2. In order to implement our application, we added a new type *SignatureSha256Ibas* `SignatureType`, which represents the IBAS algorithm [13] with *SHA-256* as a hash function. The size of the *SignatureSha256Ibas* signature value is 150 bytes, 20 bytes of which is the state information w selected by the first signer, and the remaining 130 bytes consist of two equal-sized compressed elements S_i, T_i (Section 2.3).

Table 2: NDN Signature Algorithms

Value	Signature Algorithm
0	*DigestSha256*
1	*SignatureSha256WithRsa*
3	*SignatureSha256WithEcdsa*
4	*SignatureSha256Ibas*

5.4 IBAS Control Flow for Signatures

Our IBAS implementation in NDN [19] is done by extending the ndn-cxx library [18]. The ndn-cxx library uses modules called `KeyChain` and `Validator` to sign and verify *Data* packets respectively. Therefore, we added new `sign()`, `signAndAggregate()` and `verifySignature()` methods to the ndn-cxx's `KeyChain` and `Validator` modules respectively. The two modules call `IbasSigner` to execute any IBAS related calculation. The `IbasSigner` module internally uses PBC Library [17] to execute all pairing routines such as pairing and elliptic curve generation, elliptic curve addition and multiplication, and pairing computation. The implementation design is outlined in Fig.6. The library names used at each step is displayed on the left side of the figure, where "New" means *newly implemented*.

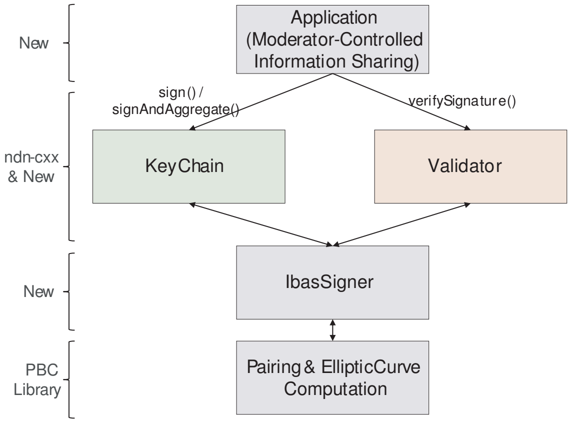
New
Application (Moderator-Controlled Information Sharing)
sign() / signAndAggregate()
verifySignature()
ndn-cxx & New
KeyChain
Validator
New
IbasSigner
PBC Library
Pairing & EllipticCurve Computation

Figure 6: IBAS Implementation Design

6. EXPERIMENTS

We compare the performance of our IBAS implementation with the built-in RSA signatures in ndn-cxx. In our experiments, only the overhead of *Data* packets (not *Interest* packets) is investigated. Technically, two alternative methods are presented: using RSA and using IBAS for signatures. For RSA signatures to be used in the PKI environment, there are two cases. (1) The subscriber shall have all of the public keys and all of the required certificates in the network. In this case, there is no need to use the `KeyLocator` field. (2) Otherwise, extra handshakes are required in order to obtain the necessary certificates using the `KeyLocater` field. The throughput of the message exchanges is measured in the case of (1).

6.1 Message Size Comparison

Table 3 shows the *Data* packet size comparison in our application, where x is the size of the message published by Alice in the scenario shown in Fig. 3. The phrases N., M.I., Con., and Sig. are abbreviations of the TLV block element names Name, MetaInfo, Content, and Signature, respectively. Since each *Data* packet has DATA-TLV of one byte and TLV-LENGTH of three bytes ahead, as '1(DATA-TLV)+3(TLV-LENGTH)' shown in Fig. 5, TLV shows this size of 4 bytes. Snippet 2 shows that *Data* packet's Signature consists of two TLV elements: `SignatureType` and `SignatureValue`. Thus, the signature sizes in Table 3 are expressed as a sum of the two numbers corresponding to `SignatureType` and `SignatureValue`. The size of SignatureInfo for RSA is variable and in this case is 82 bytes because it contains `KeyLocator`, while that for IBAS is fixed at 5 bytes. The size of each part is also shown in Fig. 5.

Table 3: *Data* packet size comparison (in bytes)

a: Publisher to Moderator (Fig. 3(3))

	TLV	N.	M.I.	Con.	Sig.	Total
RSA-2048	4	46	5	53+x	82+260	450+x
RSA-1024	4	46	5	53+x	82+132	322+x
ECDSA-521	4	46	5	53+x	82+126	314+x
IBAS	4	46	5	53+x	5+152	265+x

b: Moderator to Subscriber (Fig. 3(4))

	TLV	N.	M.I.	Con.	Sig.	Total
RSA-2048	4	74	5	452+x	87+260	882+x
RSA-1024	4	74	5	324+x	87+132	626+x
ECDSA-521	4	74	5	318+x	87+126	614+x
IBAS	4	74	5	110+x	5+152	350+x

Table 3 shows that a message overhead reduction of 45% or 60% is achieved by IBAS from RSA-1024 or RSA-2048 in this case. If Alice sends out a 6-byte message "I'm OK" ($x = 6$), the size of the corresponding *Data* packet received by Bob is $350 + 6 = 356$ bytes in the case of IBAS, or $882 + 6 = 888$ bytes in the case of RSA-2048.

The size of the elliptic curve used by this IBAS is the default 513 bits (rbits = 160, qbits = 512) of the PBC Library [17], which is implemented in 65 bytes. Although the current ndn-cxx [18] does not implement ECDSA-521 [12], Table 3 also shows its calculated overhead because it uses the closest-sized elliptic curve to the above IBAS among three ECDSA algorithms defined in RFC4754 [12]. This shows that the overhead reduction of 43% from ECDSA-521 will be achieved by IBAS.

6.2 Computational Overhead Comparison

In a high-delay, small-throughput network, the signature verification delay can be significantly reduced by using IBAS (Section 4) because there is no need to obtain further information such as certificates. However, if the network is healthy and all necessary public keys and certificates needed for RSA signature verification are present in the verifier's cache, the limiting bottleneck of communication throughput is the computational complexity of the signature generation and verification.

In order to make a computational comparison, we measured the average time for signature generation and verification. In this experiment, we chose the message size to be $x = 100$ bytes. The results are obtained by executing 100 operations and taking the average. The specifications of the CPU of the machine on which the benchmark was run are shown below. Although the machine has multiple cores, the benchmark was run on a single core.

```
vendor_id     : GenuineIntel
cpu family    : 6
model         : 58
model name    : Intel(R) Core(TM) i7-3520M CPU @ 2.90 GHz
cpu MHz       : 1,499.730
cache size    : 4,096 KB
cpu cores     : 2
bogomips      : 5,786.94
clflush size  : 64
address sizes: 36 bits physical, 48 bits virtual
```

Thus far, we have largely ignored the encryption strength of IBAS as compared with other signatures, and the size of the elliptic curve used by IBAS is set to the default 513 bits (rbits=160, qbits=512) in the PBC Library [17], which we hereinafter refer to as IBAS (512,160). Table 4 shows the computational costs of RSA signatures and those of ECDSA signatures. According to RFC5656 [25], RSA-1024 is comparable to the ECDSA of the key size from 160 to 223 bits, and RSA-2048 is comparable to the ECDSA of the key size from 224 to 255 bits. Another investigation on the computational costs by M. Yasuda et al[27] shows that RSA-1024 and RSA-2048 are comparable to ECDSA-133 and ECDSA-195, respectively. For the second IBAS implementation IBAS(224,112), we choose a conservative 225-bit key created by genaparam with rbits = 112 and qbits = 224 in the PBC Library [17].

Table 4: RSA vs. ECC

RSA	ECC Type 1[25] (NIST 800-57)	ECC Type 1[27] (M. Yasuda et al.)
1024	160-223	133
2048	224-255	195
2671		224
3072	256-383	
3241		247
7680	384-511	
15360	512+	

Table 5 show the computation costs of IBAS, ECDSA, and RSA for various key sizes. In RSA, signature generation is very expensive compared to verification. Therefore, the publishing and moderating time is longer than the subscription (verification) time. The difference in processing cost for ECDSA between signature generation and verification is not so large. This is why the processing time of the subscriber is equal to almost twice that of the publisher. Precisely speaking, the verification costs per signature of ECDSA-256 and ECDSA-384 are observed to be 1.21 and

1.25 times larger, respectively, than the signature generation costs. The ECDSA cost of the moderator is approximately an extra 2 ms consumed by processing other than the signing and verifying processes.

In IBAS, verifying a signature requires executing an expensive pairing function three times (Equation (1) in Section 2.3). Thus, Table 5 shows that the time required for subscription (verification) and moderating is longer than that for publishing (generation), because moderating includes verification.

Table 5: Signature processing times (ms)

	Publisher Alice	Moderator Government	Subscriber Bob
IBAS(512,160)	7.53	21.5	21.5
IBAS(224,112)	2.46	4.95	4.10
ECDSA-384	2.15	7.04	5.38
ECDSA-256	1.05	4.33	2.55
RSA-4096	10.2	10.4	0.446
RSA-2048	2.08	2.04	0.307
RSA-1024	0.738	0.841	0.246

Although the time required for verification at Bob by RSA-4096 does not increase significantly from that of RSA-1024, the generation time increases significantly and the signature processing time at the publisher and the moderator becomes more than 10 ms for RSA-4096. The processing time at Alice (publisher) and Government (moderator) by ECDSA-384 is less than that for RSA-4096. For signatures stronger than RSA-4096, we should use ECDSA instead of RSA. The encryption strength of IBAS(224, 112) may be comparable to RSA-2048 according to Table 4. Assuming that IBAS(224,112), ECDSA-224, and RSA-2048 have the same encryption strength, we can investigate the throughput comparisons. Since the processing time at the moderator is the longest and the bottleneck of these three methods, we calculated the throughput of communications via the moderator. According to Table 5, the throughputs of the moderator are is 490, 231, and 202 packets/s for RSA-2048, ECDSA-256, and IBAS(224,112), respectively. The performance of IBAS(224,112) is similar to that of ECDSA-256 but is 2.4 times slower than that of RSA-2048. In order to achieve a throughput of 1Mbps by IBAS(224,112), the message size x should be larger than 619bytes. Since the maximum data size of IEEE802.3 (802.2LLC) is 1497, IBAS(224,112) can easily provide a video streaming service at 1 Mbps.

The above results do not take into account the time required for placing *Data* packets into *Face* and processing *Interest*s. In order to obtain the average round trip time (RTT) between Bob and Alice in Fig. 3 in a delay-free network (i.e., the time for placing *Interest* and *Data* into *Face* is included, but the network has no delay), we set up the configurations to reflect the packet relay model presented in the second paragraph in Section 5. The RTT is measured 5 times by running the three different NDN clients (Alice (publisher), Government (moderator), and Bob (subscriber)) on one physical machine. The average RTT of 100-byte messages by IBAS(224,112) is 37.6ms (standard deviation: 2.8ms), whereas that of RSA-2048 is 22.7ms (standard deviation: 1.6ms). The latency of IBAS is 1.7 times larger than that of RSA-2048. In real life, the network has latency; if we add 100ms (the average worldwide RTT to Google[14]) to both numbers, the difference may be negligible.

7. DISCUSSION AND FUTURE RESEARCH

After a disaster, even when Bob is evacuated to a shelter with no internet connectivity but with a Wi-Fi network, he can immediately confirm the safety of Alice using his smartphone via ICN-based MIS using IBAS if the corresponding moderator and Alice happen to be in the same shelter. Furthermore, Bob even knows the safety of his mother via this moderator if his brother John carries his smartphone with this information into the shelter.

IBAS can be used in other applications that require sequential signing. For example, the registration certificate of our real estate lists the current owner as well as all the previous owners with deletion marks. In this case, a modified registration certificate is signed by a new aggregated signature created from this modified registration certificate, the previous aggregated signature, and the identity of the new owner.

The following are important issues to solve in the future.

Smartphones of Different KGCs in a Shelter

Smartphones in a shelter may use different mobile carriers. In this case, the KGCs may be different, and it is difficult to set up communications among these smartphones. However, when a disaster occurs, a new Incident Command System (ICS) is launched as part of the emergency response procedure [9]. Thus, one solution is to introduce a special KGC managed by ICS, which relays other KGCs. The implementation should be investigated in detail in the future.

IBAS Key Size

The encryption strength of IBAS is beyond the scope of the present paper. The size of the elliptic curve used by IBAS(224,122) is within the range of NIST 800-57, which is equivalent to RSA-2048. However, this may not mean that the encryption strength is almost the same as that of RSA-2048. Careful investigations on the encryption strength of IBAS must be performed in the future for comparison with RSA-2048.

Hashing onto Elliptic Curves

The IBAS signature scheme [13] uses hash functions $H_1, H_2 : \{0,1\}^* \rightarrow \mathbb{G}_1$, where $\mathbb{G}_1$ is a Gap Diffie-Hellman group. Building a secure hash function with such a property is not trivial. Thus, for ease of implementation, we simply used the Crypto++ [8] library's SHA-256 hash function and converted the resulting hash value into the $\mathbb{G}_1$ field. However, note that a better hash function can be built by using the *MapToGroup* method described in [5].

Distributing Secret Parameters and Key Revocation

In our application, we save the secret key of each participant in local storage beforehand, as in the Subscriber Identity Module (SIM) card according to the GSM [10]. The `IbasSigner` module then loads the keys from storage when signing packets.

Building an efficient key revocation system is difficult in all IBC systems. In the proposed application, we do not implement any key revocation mechanism.

Other Issues

There are other issues to be solved for ICN-based MIS.

- A moderator can delegate the message relaying function to the administrator if he wishes because of the processing power of his terminal or some other temporal reasons. If the KGC is the administrator, he can be a delegate of any moderator because KGC has the secret keys of all the peers. This is another advantage of using the identity-based signature to construct ICN-based MIS.
- Currently encryption is not implemented. Since the BF-IBE scheme uses the same bilinear pairing and has similar system parameters as IBAS, we believe that the IBAS scheme can be used with BF-IBE to provide fully functional IBC.
- The significant issue in routing is that messages can be delivered from peers to their moderator even if global connections are lost.
- Although the identity names *publisher* and *subscriber* are used, the proposed application is not a pub/sub system and its communication is driven by *Interest* packets. However, this NDN's default pull-based communication model can also be extended to have push-based multicast capability (pub/sub), as demonstrated in COPSS [6].

8. CONCLUSION

We developed the ICN-based MIS, which is the first working application for Identity-Based Aggregate Signatures. The ICN-based MIS is easy to implement in the information centric networking (ICN) framework but is difficult or inefficient to implement in traditional Internet application frameworks. The current IBAS extension for ndn-cxx [18] is not optimized, and the throughput is slower than expected. Thus, the source code is openly available through GitHub [19] for use in further research on this scheme.

Moreover, we have shown that using IBAS in NDN significantly reduces the overhead of the public key and signature. Since the verification of a *Data* packet is performed without requesting additional information, it is especially appealing in our example application of a disaster, where the network delay is large, and the safety confirmation messages are small. Moreover, based on experiments using our IBAS implementation, we estimated the throughput and latency of our application in a real healthy network. Future research includes finding and implementing proper solutions for the workarounds described in Section 7. In particular, finding a proper IBAS key size is indispensable for our IBAS application to operate in the real world.

Acknowledgements

This work has been supported by the GreenICN project (GreenICN: Architecture and Applications of Green Information Centric Networking), a research project supported jointly by the European Commission under its 7th Framework Program (contract no. 608518) and the National Institute of Information and Communications Technology (NICT) in Japan (contract no. 167).

9. REFERENCES

[1] 3GPP. Alphabets and language-specific information. Technical Specification Group Terminals 23.038i version 2.0.0, June 1996.

[2] D. Barr. Majordomo. `http://www.greatcircle.com/majordomo/FAQ.html`. Accessed: 2015-05-04.

[3] C. Bian, Z. Zhu, E. Uzun, and L. Zhang. Deploying key management on ndn testbed. Technical Report, UCLA, Peking University and PARC, 2013.

[4] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Advances in cryptology–EUROCRYPT 2003*, pages 416–432. Springer, 2003.

[5] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In *Advances in Cryptology –ASIACRYPT 2001*, pages 514–532. Springer, 2001.

[6] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. Ramakrishnan. Copss: An efficient content oriented publish/subscribe system. In *2011 Seventh ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, pages 99–110. IEEE, 2011.

[7] J. C. Choon and J. H. Cheon. An identity-based signature from gap diffie-hellman groups. In *Public key cryptography – PKC 2003*, pages 18–30. Springer, 2002.

[8] W. Dai. Crypto++ library, version: 5.6.1. `http://www.cryptopp.com/`. Accessed: 2015-05-04.

[9] T. Deal, C. Mills, and M. Deal. *All Hazard Field Guide: A Responder's Handbook Using the National Incident Management System's Incident Command System*. Amazon Services International, Inc., 2011.

[10] ETSI. Subscriber identity modules, functional characteristics. Recommendation GSM 02.17 version 3.2.9 (Release 92, Phase 1), February 1992.

[11] Facebook. Controlling what you see in news feed. `https://www.facebook.com/help/335291769884272/`. Accessed: 2015-05-04.

[12] D. Fu and J. Solinas. Ike and ikev2 authentication using the elliptic curve digital signature algorithm (ecdsa). RFC 4754, January 2007.

[13] C. Gentry and Z. Ramzan. Identity-based aggregate signatures. In *Public Key Cryptography-PKC 2006*, pages 257–273. Springer, 2006.

[14] I. Grigorik. Latency: The new web performance bottleneck. `https://www.igvita.com/2012/07/19/latency-the-new-web-performance-bottleneck/`, July 2012. Accessed: 2015.02.02.

[15] M. R. Horton. Standard for interchange of usenet messages. RFC 850, June 1983.

[16] L-Soft international. Early history of listserv. `http://www.lsoft.com/products/listserv-history.asp`. Accessed: 2015-05-04.

[17] B. Lynn. The pairing-based cryptography library, version: 0.5.14. `http://crypto.stanford.edu/pbc/`. Accessed: 2015-05-04.

[18] S. Mastorakis. Ndn c++ library with experimental extensions. `https://github.com/named-data/ndn-cxx`. Accessed: 2014-12-22.

[19] B. Namsraijav. Ndn-cxx fork with ibas support. `https://github.com/byambajav/ndn-ibas/`. Accessed: 2015-05-17.

[20] NDN Project Team. Ndn specification documentation. `http://named-data.net/wp-content/uploads/2013/11/packetformat.pdf`. Accessed: 2015-08-05.

[21] NDN Project Team. Type-length-value (tlv) encoding. `http://named-data.net/doc/ndn-tlv/tlv.html`. Accessed: 2015-08-04.

[22] T. Ogawara, Y. Kawahara, and T. Asami. Disaster-tolerant authentication system for ndn using hierarchical id-based encryption. In *2013 21st IEEE International Conference on Network Protocols (ICNP)*, pages 1–2, 2013.

[23] J. Seedorf, B. Gill, D. Kutscher, B. Schiller, and D. Kohlweyer. Demo overview: Fully decentralised authentication scheme for icn in disaster scenarios (demonstration on mobile terminals). In *Proceedings of the 1st international conference on Information-centric networking*. ACM, 2014.

[24] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO 84 on Advances in cryptology*, pages 47–53. LNCS Springer, August 1985.

[25] D. Stebila and J. Green. Elliptic curve algorithm integration in the secure shell transport layer. RFC 5656, December 2009.

[26] T. Yagyu and S. Maeda. Demo overview: reliable contents retrieval in fragmented icns for disaster scenario. In *Proceedings of the 1st international conference on Information-centric networking*, pages 193–194. ACM, 2014.

[27] M. Yasuda, T. Shimoyama, J. Kogure, and T. Izu. On the strength comparison of the ecdlp and the ifp. In *Proceedings of the 8th International Conference on Security and Cryptography for Networks*, pages 302–325. 2012.

[28] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, et al. Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73, 2014.

[29] X. Zhang, K. Chang, H. Xiong, Y. Wen, G. Shi, and G. Wang. Towards name-based trust and security for content-centric network. In *2011 19th IEEE International Conference on Network Protocols (ICNP)*, pages 1–6. IEEE, 2011.