Attribute-based Access Control for ICN Naming Scheme

Bing Li, Dijiang Huang, *Senior Member, IEEE*, Zhijie Wang, and Yan Zhu, *Member, IEEE*

Abstract—Information Centric Networking (ICN) is a new network architecture that aims to overcome the weakness of existing IP-based networking architecture. Instead of establishing a connection between the communicating hosts, ICN focuses on the content, i.e., data, transmitted in network. Content copies in ICN can be cached at different locations. The content is out of its owner's control once it is published. Thus, enforcing access control policies on distributed content copies is crucial in ICN. Attribute-Based Encryption (ABE) is a feasible approach to enforce such control mechanisms in this environment. However, applying ABE in ICN faces two challenges: from management perspective, it is complicated to manage attributes in distributed manners; from privacy protection perspective, unlike in traditional networks, the enforced content access policies are public to all the ICN users. Thus, it is desirable that unauthorized content viewers are not able to retrieve the access policy. To this end, a privacy-preserving access control scheme for ICN and its corresponding attribute management solution are presented in this paper. The proposed approach is compatible with existing flat name based ICN architectures.

Index Terms—Privacy, naming, information centric networking, access control, attribute management

1 Introduction

In traditional networking schemes, if a network entity wants to access some information content, it has to locate and connect to the server that provides such service following network routing protocols. As a result, the information is tightly associated with the location of the server. The entire network is centered around the connections between content consumers and content providers, making connection status an important factor to the network.

Witnessed by the fact that most of the network traffic is video sharing [1], various ICN architectures [2], [3], [4], [5], [6] are proposed. In ICN architecture, the focus is shifted from consumer-server connections to consumer-content connections. Thus, instead of identifying the content owner's address, the network changes to identify authentic content copies. In this way, the consumers do not need to know where the content locates, i.e., the IP address of the content owner. The content name is sufficient to direct the consumer to a content copy. Content owners publish the content, which can be copied and stored in the network using network caches [7], [8]. This design enables contents being efficiently delivered to consumers.

Though the design is efficient in retrieving content, it brings great challenges to security issues during content caching and retrieving. One of them is that traditional access control mechanisms [9] cannot be easily enforced. This is because, in ICN, content owners and consumers are not directly connected. Content owners have no control over the distributed network caches. To enforce access control to the content, several frameworks have been proposed [10], [11]. Most of them require additional authorities or secure communication channels in network to authenticate each content consumer. These schemes are sound but have too much reliance on traditional control schemes, making them inefficient in practice.

In this paper, we propose an attribute-based access control for ICN naming scheme. The proposed scheme can be divided into two levels. At the upper-level, to address the attribute management problem, we present an ontology-based attribute management solution to manage the distributed attributes in ICN network. In this scheme, attributes defined by different authorities can be synchronized more efficiently than traditional approaches. Content consumers do not need to negotiate their attribute keys when they request contents from other authorities.

At the lower-level, we propose an ABE-based naming scheme. This approach is inspired by Attribute Based Encryption (ABE) schemes [12], [13], [14]. In our approach, each network entity is assigned with a set of attributes with the help of a Trusted Third Party (TTP) according to their real identities. The access control policy is enforced according to the content names instead of the contents. Moreover, privacy-preservation is provided for the content access policies. This feature can greatly improve the privacy protection on ICN data when they are distributed in the public domain. In this way, a user is able to identify its eligibility of the accessed contents through the encrypted names before actually accessing the data content. To further support the use of ontology in attribute management, the proposed scheme enables comparison between attributes, which gives the capability to rank attributes and associate different privileges accordingly. In summary, the contributions of our work can be listed as follows:

- *B. Li, D. Huang, and Z. Wang are with the School of Computing Informatics and Decisions Systems Engineering, Arizona State University, Tempe, AZ 85281. E-mail: {Bing.Li.4, Dijiang.Huang, zhijie.wang}@asu.edu.*
- *Y. Zhu is with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China. E-mail: zhuyan@ustb.edu.cn.*

Manuscript received 11 May 2015; revised 7 Feb. 2016; accepted 19 Feb. 2016. Date of publication 5 Apr. 2016; date of current version 9 Mar. 2018.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TDSC.2016.2550437

- It provides ontology-based attribute management, which greatly reduces the cost for attribute management in distributed deployment. The proposed management scheme supports flexible attribute combination operations in access control policies;
- It enables attribute rankings and access privilege management, making it flexible to construct a data access policy in real-world scenario. The content access policy is confidentially preserved. Ineligible consumers cannot derive the data access policies even if they collude together;
- It proposes a naming scheme for ICN network which combines the flexible attribute management solution with the privacy preserving access policy;

It significantly reduces the computation and communication overhead for a potential consumer to determine his eligibility to access the content.

The remainder of this paper is organized as follows. Section 2 reviews related work on ICN and its security. Section 3 presents the system models and preliminaries. We present the ontology-based attribute management scheme in Section 4 and detailed description of the naming scheme in Section 5. Analysis and performance evaluation of the proposed solution is provided in Section 6 and our work is concluded in Section 7.

2 Related Work

In this paper, we propose an ABE-based scheme to enforce a secure access control mechanism in ICN network. Before introducing details of our approach, we first present related research results on ICN and ABE.

2.1 ICN Solutions

Several ICN architectures have been proposed in the past years. CBCB [2] runs on application layer. It uses publish/subscribe scheme to publish contents. Each consumer broadcasts its interest in the form of attribute combinations. At each router, interests associated with an interface are updated in the form of predicates. When content is transferred through the network, the content is compared with the predicates on every interface to determine through which interfaces to forward the content.

DONA [3] is deployed above IP layer. Consumers use the name resolution system to find the nearest copy of the content. The system returns with the content copy or the IP address of the content location. NetInf [4] uses a similar naming scheme as DONA. It uses multi-level Distributed Hash Table (DHT) for name resolution. PURSUIT [5] has a much different structure for retrieving the content location which involves topology information and load balance.

NDN [6] doesn't specifically define the name structure. A name in NDN consists of multiple components, each of which can be a human-readable string or a digest of the content. Content providers should guarantee the uniqueness of name components.

All these ICN solutions focus on the efficiency and security aspects of the network while access control to the content and content privacy are not well addressed. In [10], an independent access control system is introduced to support the need in ICN. This system connects to the ICN structure through a component called the Relaying Party (RP). An additional component called Access Control Provider (ACP) is in charge of creating access policies and enforcing the policies to consumers' credentials. This system incorporates access control into ICN systems but requires much more network interactions. For content privacy purposes, [15] proposes a design in which each file is divided into blocks. A block from the file is mixed with blocks from "cover" content using randomizing transformations and the results are published to the network. To recover the file, an authentic consumer needs to get more information related to the file from a secure channel. The requirement for a secure channel is difficult to meet in many ICN scenarios.

2.2 ABE Schemes

Ciphertext-Policy ABE (CP-ABE) [12] assigns each user with a set of attributes. There is one private key corresponding to each attribute value. A policy for decryption is constructed by the encryptor. This policy is transmitted together with the ciphertext, but in plaintext form. Users who do not satisfy the policy are not able to decrypt the ciphertext. This scheme can provide access control to individual messages. A content owner can specify the required attribute combinations without knowing the receivers' keys.

The reason why CP-ABE is not suitable for ICN usage is that the policy is transmitted in clear text. In traditional network, a user is authenticated before access is granted. However, once a content is published in ICN, the owner has no control on it. In this way, any network user who has access to the ciphertext can access the policy. Attackers can deduce sensitivity of the message and infer the identities of those who are involved in the message transmission.

What is needed for CP-ABE is to hide the policy into the ciphertext. For this purpose, several works [13], [16] are proposed. An attacker cannot get any information about the policy. But these solutions sacrifice efficiency to security in that any party that tries to decrypt the ciphertext will have to go through the entire decryption process which involves a heavy computation overhead.

To save computation resources for the unsatisfactory users, Huang et al. proposed a scheme [17] to expose the policy attributes step by step. In this way, the decrypter is able to stop the decryption process as soon as it fails at one step. But the price for this ability is that one additional attribute, which is the one that fails the decrypter, is exposed. Besides, this approach supports AND-gates only, which limits the flexibility of the policy.

For attribute management purpose, it is desirable to enable the comparison between attributes so that the nominal attributes can be mapped into ordinal values, e.g., $\{Nurse\} < \{Physician\}$. In [18], [19], [20], Zhu et al. proposed a encryption scheme using interval comparisons based on bilinear groups. In this paper, we adopt the idea for interval comparisons, but apply it to hidden-policy attribute based encryption algorithms.

Another attribute management concern is how to manage attributes in a distributed manner. To enable such feature, several works have been done [14], [19], [21], [22], [23] with diverse technical approaches to explore solutions for multiple authorities working securely together. The

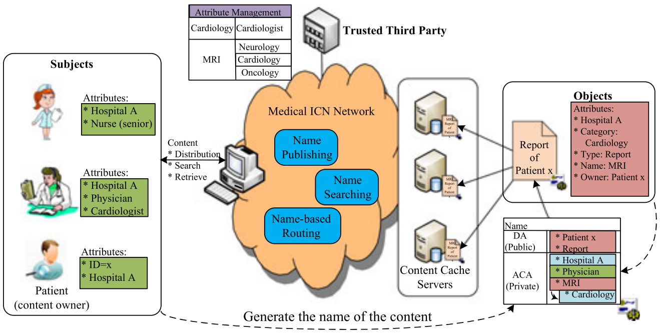


Fig. 1. Basic ICN system model.

multi-authority property of these solutions provide us references in designing our proposed scheme.

2.3 Ontologies

The standard form of defining an ontology is the Web Ontology Language (OWL) [24] which is based on the Resource Description Framework (RDF) and defines an XML based syntax for specifying all the different aspects of the ontology. We can query OWL ontologies using the SPARQL language [25] that takes into account the ontology structure while executing.

3 System and Models

In this section, we present a basic medical ICN framework, an overview of the attribute-based naming and access control model proposed in the rest of the paper, and the attack model. A preliminary introduction to bilinear map computation, which is basis to ABE schemes, is also provided.

3.1 Application Scenario

In a typical ICN system, there are three roles: content owner, content consumer and content cache. A content owner creates the content and publishes it into the network. A consumer is a network entity that requests for the content. It gets the content with the help of the ICN infrastructure. A cache is an entity that keeps a copy of the content for a period of time in its own local storage so that whenever a request for the same content arrives, it directly responds with a copy of it to the consumer. All these three network roles are exchangeable for individual network entities. That is to say, a network entity can simultaneously be a publisher, a consumer and a cache for different contents. In the following, we will use an example in medical care through out the rest of this paper to show how the proposed scheme works. As shown in Fig. 1, the content owner can be a Patient, a content consumer can be a Nurse or a Physician, and the content caches are servers storing encyrpted contents.

In an ICN network, users get content names from a Name Searching service (NSS) and use the names to get the content through a Name-based Routing (NR) system. A user gets content names from the NSS and the NR is able to retrieve the content based on the names. Details on how these two systems are implemented is out of the focus of the work. Interested readers can refer to [2], [3], and [6] for more information. Additionally, our model includes a Trusted Third Party (TTP) that sets up Attribute-Based Access Control (ABAC) and Attribute-Based Encryption (ABE) related public parameters for the network. It also helps assign and manage attributes to entities. A formal definition of TTP is:

Definition 1. *A Trusted Third Party (TTP) is an independent party that is trusted by all the network participants, which creates and manages global security parameters of the system.*

In the proposed scheme, every network entity is associated with a unique identifier (UID) and a set of attributes. UID itself can be treated as a special attribute. Attributes (other than UIDs) can be defined and managed by any entity in network. This entity is denoted as the authority of an attribute. As in this example, the attributes include: $\{HospitalA, Nurse, Physician, Cardiologist, MRI\}$. In our network model, multiple attribute authorities (AAs) are able to exist at the same time. An AA is defined as:

Definition 2. *An Attribute Authority (AA) is a semi-trusted network participant that creates and manages a set of attributes shared in the network. It is only trusted by the participants who have been assigned at least one attribute by this AA.*

Thus, not only all the network users are organized in a distributed manner, the attribute authorities are also distributed. This property is supported by the specially-designed naming scheme in Section 5. Each of the authorities is in charge of an independent set of attributes. They rely on TTP and work closely with the TTP to establish the secure components corresponding to the attributes and assign attribute keys to network participants. The entire network model consists of one TTP and multiple AAs.

3.2 Attribute-based Naming and Access Control

Attributes in an ICN network can be categorized into subject attributes and object attributes. As shown in Fig. 1, attributes in green are subject attributes while the red attributes are object attributes of the report. When they are used in ICN, there are some relations between the subject attributes and object attributes. For example, $\{Cardiology\}$ and $\{Cardiologist\}$ are a subject attribute and an object attribute, respectively. They can be treated as equal since a cardiologist always works on cardiology. Another example is $\{MRI\}$. As a useful tool, several medical subjects make use of MRI for diagnosis, such as neurology, cardiology, and oncology. To model such relationship, we define $\{Neurology, Cardiology, Oncology\}$ as sub-attributes of $\{MRI\}$. The proposed ontology-based attribute management approach is able to handle such attribute equivalence and attribute hierarchy relations. When a content owner publishes the content, he decides which attributes are used for access control and which are for content search and content description. We denote them as Access Control Attributes (ACAs) and Descriptive Attributes (DAs), respectively. As in the example of Fig. 1, $\{Hospital\ A, Physician, MRI, Cardiology\}$ are used as ACAs. $\{Patient\ x, Report\}$ are used as DAs. Thus, network entities only see that this content is a report of Patient x as the DAs are publicly search-able. The decision on ACA/DA classification is crucial to the privacy of the protected content and it's up to the content owner to make such decision.

For a given value space N_A of attributes and a namespace N_N of content names in ICN network, an Attribute-based Naming scheme is defined as:

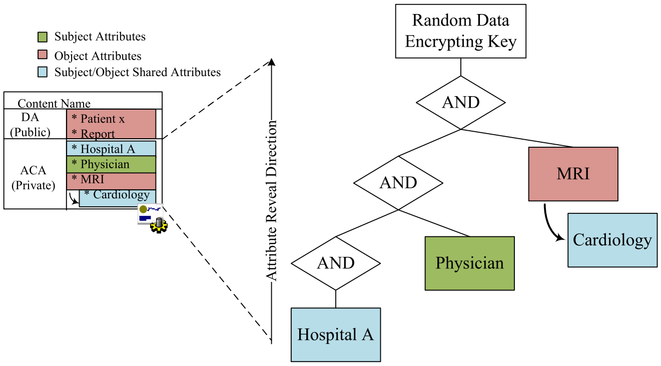


Fig. 2. Creating a content name.

Definition 3. *Given a set of ACAs $S_A \subseteq N_A$, a mapping scheme $F : \mathbb{N}_A \rightarrow \mathbb{N}_N$ for any ACA $a \in S_A$ is an attribute-based naming scheme for N_A and N_N, denoted as $F(N_A, N_N)$.*

3.3 Comparable Attributes and Attribute Rankings

In addition to the above-mentioned attribute setups, we also enable comparison between attributes. To illustrate this property, we use the example policy in 3.1, $\{A\}\, AND\, \{\{Physician\}\, OR\, \{Nurse\}\}$. If we want to modify the policy as: all the staff working at hospital A that rank higher than nurse are allowed to access the file, then in the traditional way, we need to enumerate all the attributes that are allowed and construct a very complex policy as $\{A\}\, AND\, \{\{Physician\}\, OR\, \{Nurse\}\, OR \ldots\}$. However, if we set up a relationship between $Physician$ and $Nurse$ as $Physician > Nurse$, meaning that a $Physician$ attribute includes all the privileges of a $Nurse$ attribute but with more that are not possessed by $Nurse$, then the original policy can be reduced. Suppose we have established such comparison relationship with all the related occupation roles, then $\{A\}\, AND\, \{\{Physician\}\, OR\, \{Nurse\}\, OR \ldots\}$ can be reduced to $\{A\}\, AND\, \{Nurse\}$, which is much easier to manage.

An Illustrative Example

In the example of Fig. 1, there are three subjects: a Nurse, a Physician, and a Patient. Their attributes are as shown in the figure. The patient publishes his MRI report in the network as the content. He, as the content owner, specifies an access policy as shown in Fig. 2 for the MRI report. Its object attributes are listed in Fig. 1. The content name is created following the procedure in Fig. 3, which will be further illustrated in Section 5. When the nurse tries to access this content, she can successfully use her $\{Hospital\ A\}$ attribute to decrypt the first node but will get stuck at $\{Physician\}$, meaning this content is not intended for her. When the Physician accesses the content, she can successfully decrypt the entire decryption process from the leaf to the root level-by-level to reveal the random data encrypting key. Here, $\{MRI\}$ is substituted with $\{Cardiology\}$ since $\{Cardiology\}$ is a sub-attribute. This is shown with the arrow in Fig. 2. Also, $\{Cardiology\}$ equals to $\{Cardiologist\}$ in this case. Then, the Physician uses the NR system to get the nearest copy of the content and uses the random data encrypting key derived from the name to decrypt the MRI report.

3.4 Attack Model

In the following, we assume that the attackers have two goals in compromising the ICN access control scheme: (1) acquiring unauthorized privilege to the data; (2) retrieving constitutional information of access policies to gain more information about the content, the owner, and the consumers. The information includes but is not limited to the identity of the owner or consumers, the sensitivity of the content and the potential value of data in the content. For the first goal, the attackers have to break the confidentiality mechanism of the protected data. Possible methods include collusion attacks and vulnerability exploitation. For the second goal, attackers need to analyze the proposed ABE-based scheme to identify possible ways to reveal the policy.

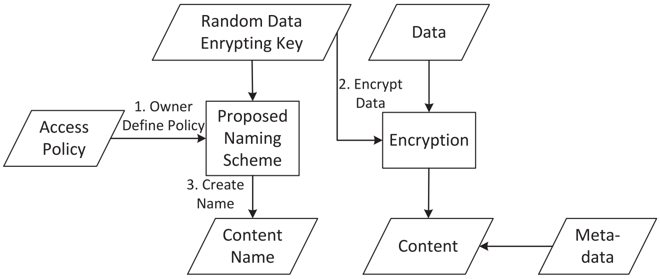


Fig. 3. Creating a content.

3.5 Preliminaries of Bilinear Map

The foundation of ABE-type algorithms is pairing computation. In this paper, we adopt the design from [19] in terms of algebraic structure. Suppose we have two groups: an additive group G_0 and a multiplicative group G_1 with a same order $n = sp'q'$, where p' and q' are two large prime numbers. We define a bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$. This map has three properties:

- Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$, for any $P, Q \in \mathbb{G}_0$ and $a, b \in \mathbb{Z}_p$;
- Nondegeneracy: $e(g, h) \neq 1$, where g and h are generators of $\mathbb{G}_0$;
- Efficiency: Computing the pairing can be efficiently achieved.

4 Ontology-Based Attribute Management

In this section, we describe the ontology-based attribute management scheme in details. This scheme is suitable for managing attribute names and values for both ACAs and DAs in a distributed manner, which is very suitable for ICN architecture.

Ontology defines various classes of users, data properties and object properties that are part of the attribute schema. Data properties are the attributes of the users and object properties are attributes of user classes that relate a user object to objects of other class. We can query Web Ontology Language (OWL) [24] ontologies using the SPARQL language [25] that takes into account the ontology structure while executing. Information about different attributes can be merged and integrated. This aggregation is useful to determine facts which cannot be acquired from any individual parties in decentralized environment.

In the ICN model (Section 3.1), we need a scheme for negotiating the attributes to be used by the users to encrypt and decrypt data, so that these processes can occur coherently. The same attribute can have different names while having the same meaning in real life like synonyms. Sometimes we also need to restrict the possible values for a particular attribute to a realistic range.

In our design, the TTP specifies an Attribute Ontology that defines the set of attributes that the users can use. Users composite the access policies with these attributes for the content they publish over the network. Our design uses the following approach that distinguishes it from the current systems.

4.1 Attribute Equivalence

The main advantage of using Ontologies to define attributes is that it allows us to declare equivalent attributes. For example, if we declare attributes (or properties) A and A′ as equivalent, some users can use the property name A for defining their policy and others can use A′ for their policy. The ontology ensures that both properties map to the same attribute, i.e., the same mathematical elements in ABE algorithm. A formal definition for such a equivalence is:

$$A \equiv A' \textbf{ iff } I_A = I_{A'}, k_A = k_{A'}, and\ h_A = h_{A'} \in Z^*_{n'}.$$

Here, I, k, h are mathematical elements used in the proposed naming scheme as in Section 5.

Equivalent properties can be specified in an OWL document using the *owl:equivalentProperty* element. We can define two attributes *attribute1* and *attribute2* as data type properties. In any one of these attributes we can declare the equivalence by adding the *owl:equivalentProperty* element in the property definition [24]. Once the equivalent set of attributes is defined, the users can use either of the attributes from the set for sharing as well as accessing data since all of them map to the same attribute. For example, if we have equivalent attributes $Faculty \equiv Teacher$ and they map to a unique triplet of $\{I, k, h\}$, then this unique triplet can be used to encrypt the data using the ABE based naming scheme.

4.2 Attribute Hierarchy

Another advantage of using ontologies for defining attributes is we can define a hierarchy of attributes. One attribute can be defined as a sub-attribute of another. For example, in a hospital, a user can have the attribute $Employee$. We can define an attribute $Nurse$ as a sub-attribute of $Employee$. This means that a “Nurse” is also an “Employee” and he can have more properties than other “Employees”. This is done by using the *rdfs:subPropertyOf* element in the OWL specification [24]. The advantage is that if the access policies specifies that all users having attribute $Employee$ can access a particular data, the attribute hierarchy specified in the ontology will also allow a $Nurse$ to get it, but any other “Employee” cannot access the data that requires the user to be a “Nurse” to access it. In Section 5.3, an attribute ranking capacity is enabled through the proposed naming scheme. The connection and difference between this example and the one in Section 5.3 is that:

- In this example, $Nurse$ and $Physician$ are two values in the category of $Employee$, or $Nurse/Physician \in Employee$;
- In Section 5.3, $Nurse$ has less privileges than $Physician$, or $Nurse < Physician$.

Here we can treat $Employee$ as a set of some ordinal attributes.

In fact, with the Attribute Hierarchy, a user is able to define the set relationship between attributes. In Section 5.3, the users can further define the ordinal relationship among all the attributes within a set.

4.3 Distributed Policy Specification

Since our design assumes multiple trusted attribute authorities, we need an initial negotiation between these authorities to decide the structure of attributes and properties to be specified in the attribute ontology. This can be achieved with the help of the TTP. Once this agreement is established, the users can access this ontology from any of the attribute authorities and design the access policy for their shared data. This policy can then be translated to an equivalent ABE compatible form to be used for encrypting the contents. This approach allows the access policies to be generated in a distributed manner by the users themselves. As long as these access policies satisfy the established ontology, they can be used over the entire network.

5 ABE-BASED ICN NAMING SCHEME

In this section, we illustrate the detailed design for the proposed ABE-based naming scheme in ICN network, which is based on our previous work [17], [26].

5.1 Creating a Content

Initially, the TTP sets up global parameters for the entire network. Then, any entity in network can create attributes and assign them to other entities. Detailed process on how attributes are distributed is out of the scope of this work. Interested reader can refer to allocation problem solutions such as [27]. Once the attributes are assigned, entities are able to create contents. As shown in Fig. 3, when an entity publishes a file, as the content owner, it creates an access policy for the content. The policy is represented as a combination of related attributes with AND and OR gates. For example, if a content owner wants to create a record accessible only to physicians and nurses working at hospital A, the policy can be $\{A\}\, AND \{\{Physician\}\, OR \{Nurse\}\}$. In this way, the owner does not need to know explicitly who should access the content. All he needs is to identify the attributes and the combination so that as long as a consumer satisfies the policy, it is able to access the content. Any entity who does not satisfy the policy will not be able to access the file in this content.

After creating the policy, the owner generates a random data encrypting key and uses it to encrypt the file. The encryption result is set as the data part of the content item. The meta-data includes public parameters to decrypt the data and data integrity related information. Then the owner creates a name for the content. He uses the proposed scheme to encrypt the random key under the policy he has specified. The result is used as the content name. Here we need to emphasize that the generated name hides the content access policies so that no one can get the entire policy from the name.

A consumer who needs this file can get the content by its name. Before he gets the content, he uses his assigned attributes to decrypt the name. If his attributes satisfy the hidden policy in the name, he can get the random data-encrypting key protected in the name. The data of the content then can be decrypted using the random key to get the original file. If a consumer cannot successfully decrypt the content name, it implies the consumer is not allowed to access the original file. Thus, even if he downloads the content, he still does not have the random data encyrpting key to decrypt the content.

5.2 ABE-based Naming Scheme

In this section, we use a composite order group G_0 with an order $n = p^2q^2$, where p and q are two large prime numbers. In other words, we fix the composite value s in Section 3.5 as equal to pq. We also choose two subgroups G_s and G_t of G_0 such that $s = pq$, $t = pq$, and G_s is orthogonal to G_t. We deliberately choose such composite-order group configuration mainly because the proposed scheme is designed to support attribute rankings in G_s. It follows RSA conditions to enforce one-direction deduction between attribute values. This is why the value of s and t are set to be products of two large prime numbers. Details of such process will be illustrated in Section 5.3.

Attributes of an entity can be any value in strings. In our scheme, each attribute string A_i corresponds to a triplet (I_i, k_i, h_i), where $I_i, k_i, h_i \in \mathbb{Z}^*_{n'}$. S_i and T_i are assigned by the TTP. The mapping from a string to such a three-tuple is determined by the authority of attribute A_i. An access policy can be expressed in Disjunctive Normal Form (DNF) of attributes. In each conjunctive clause of the DNF, the sequence of attributes is determined by the encryptor. The sequence of encrypting a conjunctive clause (encryption sequence) is opposite to the decryption sequence. We define a public attribute A_{Pub} in the scheme. Unlike other attributes, A_{Pub} is associated with a triplet $(S_{Pub}, T_{Pub}, I_{Pub})$. For each conjunctive clause, the encryptor adds A_{Pub} at the end of the encryption sequence. In other words, the special attribute A_{Pub} is always the last attribute in encryption and the first attribute in decryption.

In the proposed scheme, a *GlobalSetup* algorithm is run by the TTP to generate global parameters for the system. For each node joining in the network, the TTP runs *NodeJoin* algorithm once to generate a unique secret for the node. For each attribute, the authority in charge runs an *AuthoritySetup* algorithm to generate secrets associated with that attribute. Besides, this naming scheme includes other three basic algorithms: *KeyGen*, *Encrypt*, and *Decrypt*. Once set up, the authority of an attribute runs *KeyGen* for each node carrying this attribute to allocate the inherent attribute secrets. *Encrypt* and *Decrypt* are used by encryptors and decrypters respectively for message passing.

The *GlobalSetup* algorithm generates global parameters $\{\mathbb{G}_s, \mathbb{G}_t, \varphi, \psi, \varphi^\beta, e(\varphi,\psi)^\alpha, Enc_k(\cdot), Dec_k(\cdot), (P_{Pub}, S_{Pub}, T_{Pub}), ROOT\}$, and global secrets $\{\beta, g^\alpha\}$, where α and β are random values and $Enc_k(\cdot)$, $Dec_k(\cdot)$ are a pair of symmetric encryption algorithms.

Algorithm 1. GlobalSetup

```
Choose two bilinear group $\mathbb{G}_0$ and $\mathbb{G}_1$ with a composite order
  $n = p^2q^2$, where $p$ and $q$ are two large prime numbers. $g$ is the
  generator of $\mathbb{G}_0$;
Choose two subgroups $\mathbb{G}_s$ and $\mathbb{G}_t$ of $\mathbb{G}_0$ such that: the order
  of $\mathbb{G}_s$ and $\mathbb{G}_t$ are both $n' = pq$; $\mathbb{G}_s$ and $\mathbb{G}_t$ are orthogonal to
  each other;
Choose two generators $\varphi \in \mathbb{G}_s$ and $\psi \in \mathbb{G}_t$;
Choose two random values $\alpha, \beta \in \mathbb{Z}^*_{n'}$;
Define a constant $ROOT \in \mathbb{G}_1$ as identification of the secret
  message;
Choose a pair of symmetric encryption algorithms $Enc_k(\cdot)$
  and $Dec_k(\cdot)$ in $\mathbb{G}_1$;
Define a public attribute, $(S_{Pub}, T_{Pub}, I_{Pub}), S_{Pub} \in \mathbb{G}_s, T_{Pub} \in$
  $\mathbb{G}_t, I_{Pub} \in \mathbb{Z}^*_{n'}$;
The global parameters are $\{\mathbb{G}_s, \mathbb{G}_t, \varphi, \psi, \varphi^\beta, e(\varphi,\psi)^\alpha, Enc_k(\cdot),$
  $Dec_k(\cdot), (S_{Pub}, T_{Pub}, I_{Pub}), ROOT\}$, global secrets are $\{\beta, \psi^\alpha\}$.
```

The *NodeJoin* algorithm is defined as in *Algorithm 2*.

Algorithm 2. NodeJoin

```
For each node with $UID$ in network, generate a random
  number $r_{UID} \in \mathbb{Z}^*_{n'}$;
Calculate $D_{UID} = \psi^{(\alpha + r_{UID})/\beta}$;
Calculate:
  $$X_{Pub,UID} = \varphi^{r_{UID}} S_{Pub}^{r_{Pub}},$$
  $$Y_{Pub} = \varphi^{r_{Pub}},$$
  $$Z_{Pub,UID} = e(\varphi,\psi)^{r_{UID} I_{Pub}}.$$
  where $r_{Pub} \in \mathbb{Z}^*_{n'}$ is a random number for each node;
Choose a random value $P_{UID} \in \mathbb{Z}^*_{n'}$;
Assign to the node $\{D_{UID}, X_{Pub,UID}, Y_{Pub}, Z_{Pub,UID}, P_{UID}\}$.
```

Each individual authority that manages an attribute A_i will have to run *AuthoritySetup* to set up attribute secrets.

Algorithm 3. AuthoritySetup

```
For each attribute $A_i$, choose random numbers $I_i, k_i, h_i \in \mathbb{Z}^*_{n'}$;
For each attribute $A_i$, generate $S_i \in \mathbb{G}_s$ and $T_i \in \mathbb{G}_t$, where $S_i$
  $= \varphi^{h_i}$ and $T_i = \psi^{h_i}$.
```

The *KeyGen* algorithm generates the private keys corresponding to each attribute for each node holding this attribute. It is defined in *Algorithm 4*. When the node receives the keys from the authority, it checks if $L_{UID}^{P_{UID}} = T_{Pub}$ is true. If it's true, it updates P_{UID} with P_{UID}^2 and accepts the keys. This update is intended to prevent from replay attack on L_{UID}. If not true, it will discard the keys.

The *Encrypt* algorithm works following the encryption sequence of each clause, denote each attribute from I_1 to I_m, m is the number of attributes in the clause. In the example of Fig. 2, $I_1 = MRI$, $I_2 = Physician$, $I_3 = Hospital\ A$, $I_4 = A_{Pub}$, $m = 4$. Choose a random value $s \in Z_p$, set $I_0 = s$ and follow *Algorithm 5*.

The *Decrypt* algorithm works in the decryption sequence. Note that the first attribute in decryption sequence is always A_{Pub}. The decrypter follows *Algorithm 6*.

Algorithm 4. KeyGen

```
1: The authority passes $I_i$, $S_i$ and $T_i$ to TTP;
2: TTP computes and sends back to the authority:
       $X_{i,UID} = \varphi^{r_{UID}} S_i^{r_i},$
       $Y_i = \varphi^{r_i},$
       $Z_{i,UID} = e(\varphi, \psi)^{r_{UID} I_i},$
       $L_{UID} = T_{Pub}^{1/P_{UID}}.$
   where $r_i \in \mathbb{Z}_{n'}^*$ is a random number;
3: The authority assigns $X_{i,UID}$, $Y_i$, $Z_{i,UID}$, and $L_{UID}$ to the node
   together with $I_i$, $h_i$ and $k_i$.
```

Algorithm 5. Encrypt

```
1: Calculate $C = Ke(\varphi, \psi)^{\alpha s}$, $C' = \varphi^{\beta s}$ and $C'' = Enc_K(ROOT)$;
2: For each attribute $A_n$, if a triplet $(C_{1,n}, C_{2,n}, C_{3,n})$ has already
   been calculated, move to the next attribute $A_{n+1}$ and restart
   step 3 with $A_{n+1}$; else, goto step 4;
3: Choose a random number $l_n \in \mathbb{Z}_{n'}^*$;
4: Calculate:
       $C_{1,n} = \psi^{(I_{n-1} - I_n) l_n},$
       $C_{2,n} = T_n^{(I_{n-1} - I_n) l_n},$
       $C_{3,n} = (k_n l_n)^{-1}.$
   $1 \leq n \leq m$;
5: Calculate $C_{1,m+1} = \psi^{(I_m - I_{Pub})}$, $C_{2,m+1} = T_{Pub}^{(I_m - I_{Pub})}$.
```

Algorithm 6. Decrypt

```
1: Start from the public attribute $A_{Pub}$;
2: For each attribute $A_n$ that the decrypter possesses, compute:
       $\frac{Z_{n,UID_{dec}} \cdot e(X_{n,UID_{dec}}, (C_{1,n})^{k_n C_{3,n}})}{e(Y_n, (C_{2,n})^{k_n C_{3,n}})} = e(\varphi, \psi)^{r_{UID_{dec}}(I_{n-1})};$
3: If $e(\varphi, \psi)^{r_{UID_{dec}}(I_{n-1})}$ is the decrypter's private key, go to step 2
   with attribute $A_{n-1}$; elsego to step 4;
4: Calculate
       $S_k = C / (e(C', D_{UID}) / e(\varphi, \psi)^{r_{UID_{dec}}(I_{n-1})}).$
if $Dec_{S_k}(C'') == ROOT$, Success; else Failure.
```

When *Decrypt* algorithm succeeds, S_k is the random data encyrpting key embedded in C.

5.3 Attribute Rankings

The proposed ABE scheme extends capabilities of traditional ABE schemes and is able to support comparison between values of the same attribute. In real world scenario, this means, for instance, two attributes values $Physician$ and $Nurse$ of attribute *Occupation* can be compared and have the relationship $Physician > Nurse$, meaning that the $Physician$ attribute subsumes all the privileges the $Nurse$ has, but the $Nurse$ does not have any of the additional privileges the $Physician$ has. Such capability is applicable when capabilities of the lower-ranking role ($Nurse$) is a subset of that of the higher-ranking role ($Physician$). In traditional ABE solutions, each attribute value ($Physician$ and $Nurse$ in the above example) corresponds to a set of cryptographic components that are designated for that specific attribute (*Occupation* in the example) of a specific user. Components for different values of the same attribute are not related. In other words, the key components of $Physician$ are independent to those of $Nurse$. To establish ranking relations between attribute values, certain connections need to be created between the corresponding key components. Specifically, a one-direction relation between values of the same attribute is supported in the proposed scheme. It allows a higher-ranking user ($Physician$) to be able to legally derive the corresponding lower-ranking role ($Nurse$) key components for himself. However, the lower-ranking role cannot derive anything regarding the higher-ranking role.

Such capability can be achieved by deliberately assigning appropriate values in *KeyGen* algorithm. We assign h_P for $Physician$ and h_N for $Nurse$ such that $h_P = h^{\alpha_P}$, $h_N = h^{\alpha_N}$, $h \in \mathbb{Z}_{n'}^*$, and $\alpha_P < \alpha_N$. Thus, we have $S_P = \varphi^{h_P}$ and $S_N = \varphi^{h_N}$. This is different from traditional ABE scheme, where both S_P and S_N are randomly chosen. This is the connection we establish between comparable values ($Physician$ and $Nurse$) of the same attribute (*Occupation*). Recall when we define the order of $\mathbb{G}_s$, it is written as $n' = pq$, where p and q are two large prime numbers. In other words, n' is a composite number satisfying RSA algorithm requirements. If a user U_P is assigned with $S_P = \varphi^{h^{\alpha_P}}$, i.e., the key for $Physician$, it is able to calculate the corresponding key S_N for $Nurse$ as long as $\alpha_P < \alpha_N$. This can be done as:

$$S_N = \varphi^{h^{\alpha_N}} = (\varphi^{h^{\alpha_P}})^{h^{\alpha_N - \alpha_P}} = (S_P)^{h^{\alpha_N - \alpha_P}}. \tag{1}$$

This means when we assign attributes to U_P, we can choose to assign the value $h^{\alpha_N - \alpha_P}$ to the user together with S_P. Thus, when the user needs to decode some message dedicated for $Nurse$, he can easily calculate S_N following equation (1). However, if another user U_N has the attribute $Nurse$, he cannot deduce S_P following the same equation in a similar way. This is because in this case, $\alpha_P - \alpha_N < 0$. Under RSA assumption, h^{-1} cannot be efficiently computed due to the secrecy of n'. A benefit of such extension to our original scheme is that it allows the ranking relations among attributes without incurring too much workload on the TTP side. Only eligible users, $Physician$ owners in this example, can use such capability and the value $h^{\alpha_N - \alpha_P}$ is only useful to eligible users.

With such a knowledge, the TTP can assign two more values Δh and Δr to user U_P in *KeyGen* algorithms. When needed, the user can derive his key values corresponding to attribute $Nurse$ afterwards. The modified step 3 of *KeyGen* is as:

$$\begin{aligned}X_{P,UID} &= \varphi^{r_{UID}} S_P^{r_P},\\ Y_P &= \varphi^{r_P},\\ Z_{P,UID} &= e(\varphi,\psi)^{r_{UID} I_P},\\ L_{UID} &= T_{Pub}^{1/P_{UID}},\\ \Delta h &= h^{(\alpha_N-\alpha_P) r_P},\\ \Delta r &= \Delta h I_N / I_P.\end{aligned}$$

Thus, the r_{UID} for U_P's $Nurse$ attribute is changed to $r'_{UID} = r_{UID}\Delta h$. Correspondingly, we have:

$$\begin{aligned}X_{N,UID} &= (X_{P,UID})^{\Delta h} = \varphi^{r_{UID}\Delta h} S_N^{r_P} = \varphi^{r'_{UID}} S_N^{r_P},\\ Y_N &= Y_P,\\ Z_{N,UID} &= (Z_{P,UID})^{\Delta r} = (e(\varphi,\psi)^{r_{UID} I_P})^{\Delta h I_N / I_P}\\ &= e(\varphi,\psi)^{r_{UID}\Delta h I_P} = e(\varphi,\psi)^{r'_{UID} I_P},\\ L'_{UID} &= L_{UID}.\end{aligned}$$

Here, we need to point out that to make sure the values of h for two comparable attributes are the same, comparable attributes need to be managed by the same authority. This means one single authority defines the relative order between these attributes. This requirement makes sense in real-world scenario since in most cases a single authority (the hospital in this example) defines values of the same attribute (job position).

5.4 Apply ABE-based Naming Scheme in ICN

With the above Naming scheme, we can achieve the following capabilities:

- Specifying the access control policy without knowing the consumers' keys;
- Full preservation of the policy confidentiality from leaking to adversaries;
- Step-by-step attribute exposure for consumers to determine their eligibility efficiently in computation; Flexible attribute management.

Using this scheme, any entity who wants to publish data needs to create the content following the process in Fig. 3.

The owner firstly creates a random symmetric key K. Then the data to be published is encrypted using K. The resulting ciphertext C is then used to generate a metadata of C. Both the metadata and C are parts of the final content. Then the owner needs to specify an access policy P of attributes to define what an authentic consumer should satisfy. Then the owner uses this policy to encrypt K following *Encrypt* algorithm. The result is used as the content name.

In this way, the owner does not need to know individual public keys of all the potential consumers in advance, which is required in traditional methods.

6 Performance Analysis and Evaluation

In this section, we firstly evaluate the performance of the attribute management scheme. Then, the computation and communication performance for the privacy-preserving naming scheme is analyzed. The security strength of the proposed naming scheme is evaluated in the last part.

6.1 Evaluation and Analysis on the Attribute Management

In this section we create an application scenario to demonstrate the effectiveness of ontology-based attribute management. Evaluation in this section is carried out in NDN environment [6]. We implemented ndn-cxx v0.3.1 library [28] together with NDF 0.3.1 [29] on Ubuntu 12.04 systems. As mentioned in Section 2.1, users are able to define their own name structure in NDN network, which makes it quite suitable for evaluating the proposed naming scheme in this paper.

In an ICN healthcare network, there are hospitals, clinics, and medical institutions that share information and data. Each institution defines its own users and attributes. A trusted Attribute Authority (AA) is set for each institution to generate attributes for users and assign them correctly when new users join in the system. It also generates the private keys to be used for ABE operations by the users and securely transfers them to intended user(s). An institution defines its own setup like the various departments, employees, patients, etc. We assume that each institution will use its own nomenclature so that the same department can have a different name in different institutions. The users can have a hierarchy among themselves, for example Nurse < Resident < Consultant < Surgeon < Head of Department < Chief Medical Officer.

6.1.1 Medical Records Transfer

When a patient is being transferred from one hospital to another for better treatment, his medical records need to be transferred as well. Since these are highly sensitive data, they are stored in encrypted form (in our case the encryption is ABE-based) and only medical personnel having the required privileges can access and decrypt this data. Let T_E be the time taken to encrypt data and T_D be the time to decrypt it.

If no ontology has been specified by the two hospitals and no attributes are shared among them, the attributes used to encrypt the data by the first hospital would be invalid in the second hospital. Therefore, the data has to be decrypted first, and then encrypted using the attributes defined in the second hospital. If we also consider the initial encryption of the data, it accounts for two encryptions and one decryption: $T_{total} = 2*T_E + T_D$.

Now we consider using an ontology that maps the attributes of one hospital to its equivalent attributes in another hospital over the ICN network. Since the attributes are already mapped, we don't need to encrypt/decrypt the data again. Only the initial encryption is required to be performed in this case: $T_{total} = T_E$.

A comparison of the two cases is illustrated in Fig. 4. As shown in the figure, using ontology based management can save about 75 percent in time when the number of attributes is more than 6, which is significant in real life.

6.1.2 Applying User Hierarchy to Access Policy

When certain data is to be shared with all doctors, irrespective of their rankings, without ontology, the person who

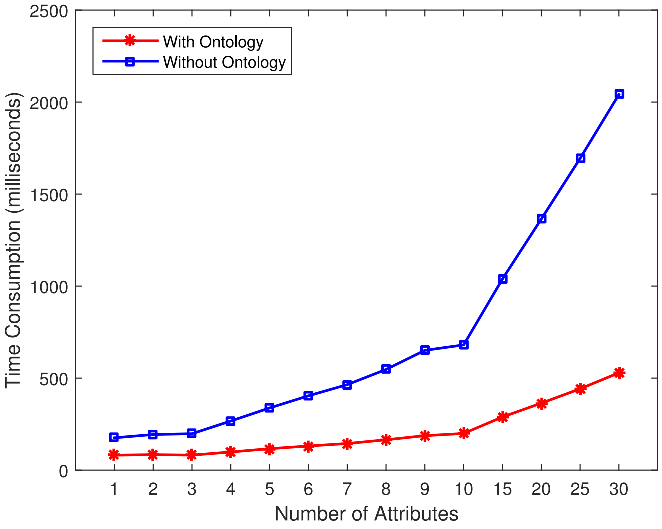


Fig. 4. Time consumption for medical records transfer.

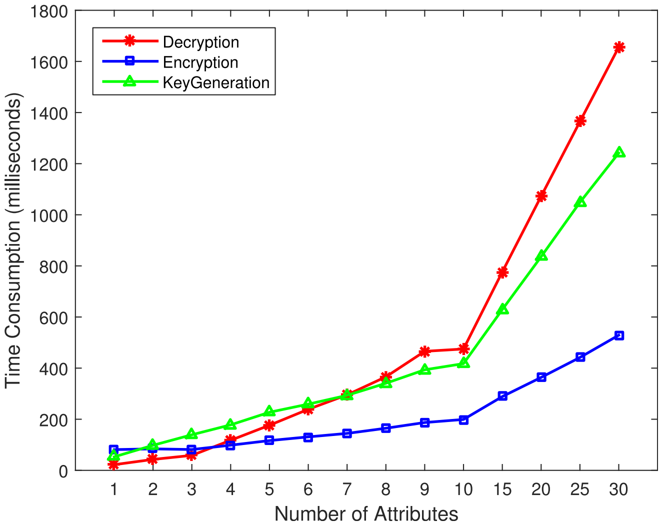


Fig. 5. Computation performance.

issues the policy will need to know all the possible ”ranks” of the doctors and then specify a DNF of all these ranks as the access policy specification. If we have an ontology where the hierarchy is specified, i.e., all the various ranks of doctors (Resident, Surgeon, etc.) are structured as sub-types of a parent type *Doctor*, the policy specifier only specifies *Doctor* as the access privilege and the ontology ensures that all the sub-types also satisfy this access requirement. In this way, if we are able to provide a proper parent type for the cases where a large combination of roles are included in the access policy, the size of the policy as well as the corresponding computation, communication, and storage cost will be greatly reduced. For such a gain, the cost is merely one additional level in the attribute hierarchy.

6.1.3 Storage Overhead and its Effect on Network Throughput

When N institutions are sharing data without using an ICN network, it results in N copies of the data, one for each institution and encrypted using its own set of attributes. Using ICN networking without the attribute management scheme can help improve the network efficiency in file sharing and content distribution, but no benefits will be provided in terms of storage overhead. With the use of ICN network in our design along with ontology-based management to specify relation between the attribute sets of all the different institutions, we reduce the storage overhead to a single copy of the encrypted data. Any eligible user from the various institutions can decrypt the encrypted data, thereby removing the need for multiple copies.

Due to the large differences in caching strategies and hardware capacities, we cannot provide a concrete network performance improvement from the reduced storage overhead. Instead, we calculate the numerical network throughput improvement using a demonstration scenario. Here, we assume that there are m domains with n entities in each domain. A file of k bits is to be shared from one single source to all the entities. The caching capacity for each network cache is m, meaning all the m copies of one single file can be cached at the same time. We also assume that the entities of each domain are evenly distributed in the network. The time needed for distributing one single copy is t seconds. When no inter-domain attribute management scheme is involved, the throughput of the network is: $\frac{\frac{m\times s}{m}}{t} = s/t$ bps. However, when the ontology-based management scheme is applied, the throughput is increased to: $\frac{\frac{s}{m}}{t} = ms/t$ bps. Such increase is due to the reduced amount of copies need to be transfered in the network.

6.2 Evaluation of the Naming Scheme

In this section, the ABE-based naming scheme is evaluated from performance aspect. We analyze its computation and communication (storage) overhead.

From computation perspective, we tested the time consumption for key generation, encryption and decryption processes. In real application, we are more concerned with the time consumption for a consumer to decrypt the content's name. This is because each content is encrypted once but decrypted by multiple users. Therefore, we also compare the decryption overhead with existing ABE solutions: CP-ABE [12], CN scheme [30], NYO scheme (the 2nd construction in [16]), YRL scheme [13] and GIE scheme [17]. The idea is to compare the number of most time-consuming operations needed in each scheme.

We use a machine with a four-core 2.80 GHz processor and 4 GB memory running Ubuntu 10.04 for experiment. PBC Library [31] is used to handle the pairing computations. We generated a type-A1 curve [32] using the parameter generating tools included in this library for the following tests. It randomly generates the prime numbers used for the curve, with a length of 512 bit for each of them. We run each operation ten times for key generation, encryption and decryption (Fig. 5). Here the policies are set to be a conjunctive clause of different number (shown in x-axis) of attributes. This is because given a number of attributes, this form requires the most time for computation. The reason why encryption consumes more time when attributes are fewer is that the cost for computing C in Algorithm 5 requires an additional pairing computation which is independent to the number of attributes. When few attributes are involved, this additional pairing takes a high portion of

TABLE 1
Time-Consumption of Different Operations (in Milliseconds)

	Pairing	Exponentiation	Multiplication	Inversion
Time	7.675	0.491	0.029	0.024

TABLE 2
Comparison of Computation Cost in Decryption

Scheme	Hidden Policy	Number of Pairings
CP-ABE	No	$2N_{invo}+1$ or 0
CN	No	$N_{all}+1$ or 0
NYO	Yes	$2N_{attr}+1$
YRL	Yes	$2N_{attr}+2$
GIE	Yes	$3N_{invo}$ or $3N_{part}$
ICN-ABE	Yes	$2N_{invo}+1$ or $2N_{part}$

the time consumption. This portion reduces as the attribute numbers grow.

In theory, the time consumption should be linear to the number of attributes involved. The curve in Fig. 5 is not perfectly linear, but it meets our expectation. There are several reasons why it is not strictly linear. Before decrypting attribute by attribute, in our program, there are some necessary steps to initialize global parameters, read files and allocate memory space. Similarly, at the end of the algorithm, we have some clean-up work, such as writing files and release memory space. Such time consumption is related to the number of attributes involved but not strictly proportional. Also, at step 4 of Decryption algorithm, there is one additional pairing operation. Thus, when the number of attributes is small, this additional operation takes more portion of the total time than when the number of attributes is large. If we further consider the possible variance introduced by system level factors, for instance the resource consumption from other processes, the variance in the figures is reasonable in practice.

For comparison purpose, we test every operation for 50 times and choose the average value as basics for our comparison. Results of our experiment (Table 1) show that pairing operation takes longer than any other operations. Therefore, our comparison metric is set to be the number of pairing operations in decryption process.

Following the above-mentioned idea, we use N_{attr} to denote the number of attributes a consumer has, N_{all} as the total number of attributes defined in the network ($N_{all} \gg N_{attr}$). The proposed naming scheme is denoted as ICN-ABE in the rest of this paper. Since the policy is publicly known in CP-ABE and CN, decrypters are able to decide what attributes to use in decryption. Therefore, for those who satisfy the policy, the time taken for decryption is proportional to the number of attributes involved, which is denoted as N_{invo}, $N_{invo} \leqslant N_{attr}$. It is obvious that inauthentic decrypters would not bother to try decryption, which is why it takes 0 in time. An inauthentic decrypter in GIE and ICN-ABE is not able to proceed with the decryption process if it cannot meet the next attribute. In this situation, we use N_{part} to denote the number of attributes that the consumer has already decrypted, where $N_{part} \leqslant N_{invo}$. The result of our test is shown in Table 2.

To evaluate the communication costs, we compare the size of the name. In PBC library [31], a data structure element_t with size of 8 bytes is used to represent an element. For our scheme, we need 24 bytes to store the network name. Compared with this name size, a content in CBCB [2] is identified by a set of attributes determined by the content owners. Thus, we can model the names as a human-readable string of an undetermined size. NDN [6] shares a similar problem with the name size. As mentioned before, DONA [3], NetInf [4] and PURSUIT [5] share the same naming scheme. Therefore, we only use the size of DONA's name for comparison. In [3], the size of the name is confined to 40 bytes in its protocol header. Thus, the network name size in our scheme is small enough to fit in existing ICN solutions.

6.3 Security Analysis

From the security perspective of the proposed solution, we analyze the performance based on the attack model in 3.4. For the first attack goal, we will provide a security theorem and give the security proof as appendix in this paper. For the second goal, we will analyze the scheme based on the algorithms.

Theorem 1. *Let G_0 and G_1 defined as in Section 5. For any adversary A, the advantage it can gain from the interaction with the security game defined in Appendix A is negligible.*

The proof for this theorem is provided in Appendix B. In the proof, it is verified that the attacker cannot break the encryption algorithm to get any data exposed. Furthermore, it is also proved that attackers cannot conduct collusion attacks onto the system.

For the second attack goal, the attacker will stop at the first attribute, A_k, that he doesn't own in the decryption process. If he can get to know this additional attribute, he must get it from step 3 in Algorithm 6. This means that the attacker possesses the secret key $Z_{i,UID}$ of the attribute A_k, which contradicts to the assumption that he does not possess this attribute.

7 Conclusion

In this paper, we proposed a comprehensive access control solution for ICN network. This solution is based on an ontology-based attribute management scheme and a privacy-preserving ABE-based naming scheme. The ontology-based scheme supports flexible attribute management with significant performance gains in terms of time consumption, storage costs, and throughput improvement. From security and privacy perspective, the ABE-based naming scheme achieves a high security level as CP-ABE, but with attribute anonymity protection for policy privacy and flexible

TABLE 3
Comparison of Ciphertext Size

Scheme	Ciphertext Size
CP-ABE	$1\mathbb{G}_1 + (2N_{ciph}+1)\mathbb{G}_0$
CN	$1\mathbb{G}_1 + (N_{all}+1)\mathbb{G}_0$
NYO	$\geqslant 1\mathbb{G}_1 + (2N_{all}+1)\mathbb{G}_0$
YRL	$1\mathbb{G}_1 + (3N_{all}+3)\mathbb{G}_0$
GIE	$N_{ciph}\mathbb{G}_1 + 3N_{ciph}\mathbb{G}_0$
ICN-ABE	$1\mathbb{G}_1 + (2N_{ciph}+4)\mathbb{G}_0 + N_{ciph}\mathbb{Z}_p$

attribute rankings. Experiments and analysis confirm the effectiveness of our schemes and design.

Appendix A ABE Security Model

As mentioned in Section 1, the proposed solution includes an upper-level ontology-based attribute management scheme and a lower-level ABE-based naming scheme. The upper-level is only focused on the relationship between attributes, which has little to do with security. In this section, we only focus on the naming scheme, which can be modeled in the form of a game between a challenger and an adversary. The challenger simulates the operations of the TTP and the attribute authorities, while the adversary tries to impersonate as a number of normal network nodes. The game consists of the following five steps:

- *Setup*. The challenger runs the *GlobalSetup* algorithm and returns to the adversary the parameters.
- *Phase 1*. The adversary can ask for an arbitrary number of user keys from the challenger. The challenger runs the *NodeJoin* algorithm for each user involved in the requests and returns the corresponding secret information. The adversary then plays in the roles of these users to request for attributes from the challenger. The challenger runs the *AuthoritySetup* algorithm to create parameters for authorities and runs the *KeyGen* algorithm to generate keys on behalf of the authorities and the TTP.
- *Challenge*. The adversary provides two messages M_0 and M_1 to the challenger together with an access policy A. A satisfies that none of the users created by the challenger has attributes satisfying A. The challenger flips a coin b and encrypts M_b using A. It sends the ciphertext back to the adversary.
- *Phase 2*. The adversary can ask for more attributes and users from the challenger. But if any single user can gain satisfactory attribute combinations for A, the challenger aborts the game.
- *Guess*. The adversary makes a guess b′ for the real value of b.

The adversary's advantage in this game can be defined as $ADV = P[b' = b] - \frac{1}{2}$. The proposed scheme is secure if for all the polynomial time adversaries, the advantage is at most negligible in the game.

Appendix B Security Proof Sketch

In this section, we provide the sketch for security proof following the structure in [12]. Before going into details of the proof, we first modify the security game described in Section A. This modification follows the same idea as in [12] and it is intended to change from differentiating two random messages M_0, M_1 to differentiating $e(\varphi, \psi)^{\alpha s_j}, e(\varphi, \psi)^{\theta_j}$ so that the generated intermediate results can be represented using the four mappings introduced in Section B.2. The goal of such modification is essentially to facilitate the following security proof. To differentiate these two games, we call the one in Section A as *Game1* and the modified game as *Game2*.

B.1 Modified Game

Game2 consists of five steps similar to *Game1*. The steps *Setup*, *Phase1*, and *Phase 2* are the same as in *Game1*. The *Challenge* step is different in that the challenger does not choose one message to construct the ciphertext C. Instead, it outputs C_j as:

$$C_j = \begin{cases} e(\varphi, \psi)^{\alpha s_j} & \text{if } b = 1 \\ e(\varphi, \psi)^{\theta_j} & \text{if } b = 0. \end{cases}$$

Here, all the θ_j are randomly chosen from $Z^*_{n'}$ following independent uniform distribution.

Suppose an adversary *adv1* in *Game1* has the advantage of ϵ, the corresponding adversary *adv2* in *Game2* can be constructed according to the following strategy:

- Forward all the messages between *adv1* and the challenger during *Setup*, *Phase1*, and *Phase 2*;
- In the *Challenge* step, *adv2* gets two messages M_0 and M_1 from *adv1* and the challenge C from the challenger. *adv2* flips a coin δ and sends $C' = M_\delta C$ to *adv1* as the challenge for *adv1* in *Game1*. *adv2* generates its guess based on the output δ' from *adv1*. If $\delta' = \delta$, then the guess is 1; otherwise, it is 0.

The advantage that *adv2* has in this game can be calculated as $\frac{\delta}{2}$.

In the following, we will show that no polynomial adversary can distinguish between $e(\varphi, \psi)^{\alpha s}$ and $e(\varphi, \psi)^{\theta}$. Therefore, no adversary can have non-negligible advantage in the security model.

B.2 Security Guarantee in the Modified Game

In this section, we follow the generic group model introduced in [33] and use a simulator to model the modified security game between the challenger and the adversary. The simulator chooses random generators $\varphi \in G_s$ and $\psi \in G_t$. It then encodes any member in G_s and G_t to a random string following two mappings: $f_0, f_1 : \mathbb{Z}_{n'} \rightarrow \{0, 1\}^{\lceil \log n' \rceil}$. It also encodes any member in G_1 to a random string in a similar way: $f_2 : \mathbb{Z}_n \rightarrow \{0, 1\}^{\lceil \log n \rceil}$. One additional mapping f_3 is used to convert elements in $\mathbb{Z}^*_{n'}$ to string representations: $f_3 : \mathbb{Z}^*_{n'} \rightarrow \{0, 1\}^{\lceil \log n' \rceil}$. The four mappings should be invertible so that the simulator and the adversary can map between the strings and the elements of corresponding algebraic structures in both directions. Four oracles are provided to the adversary by the simulator to simulate the group operations in G_s, G_t, G_1, and the pairing. Only the string representations can be applied to the oracles. The results are returned from the simulator in the string representations as well. The oracles will strictly accept inputs from the same group. The simulator plays the role as the challenger in the modified game.

- *Setup*. The simulator chooses G_s, G_t, G_1, e, φ, ψ, and random values α, β. It also defines the mappings f_0, f_1, f_2 and the four oracles mentioned above. The simulator chooses the public attribute parameters $I_{Pub} \in \mathbb{Z}^*_{n'}$, $S_{Pub} = f_0(\mu) \in G_s$, $T_{Pub} = f_1(\lambda) \in G_t$, and $ROOT \in G_1$, where λ and μ are random strings. The public parameters are G_s, G_t, $\varphi := f_0(1)$, $\psi := f_1(1)$, $\varphi^{\beta} := f_0(\beta)$, $e(\varphi, \psi)^{\alpha} := f_2(\alpha)$, $(S_{Pub}, T_{Pub}, I_{Pub})$, and $ROOT$.

TABLE 4
Query Information Accessible to the Adversary

μ	β	$r_{UID} + \mu r_{Pub,UID}$
r_{Pub}	h_i	$r_{UID} + h_i r_i$
r_i	$(I_{n-1} - I_n)h_n$	$t_n(I_{n-1} - I_n)h_n$
λ	$(\alpha + r_{UID})/\beta$	βs
h_i		
α	$r_{UID}I_{Pub}$	$r_{UID}I_i$
I_{Pub}	I_i	k_i
$(k_n t_n)^{-1}$	h_i	

- *Phase 1.* When the adversary runs *NodeJoin* for a new user with *UID*, the simulator generates a random number $r_{UID} \in \mathbb{Z}^*_{n'}$. It returns to the adversary with $D_{UID} = f_1((\alpha + r_{UID})/\beta)$, $X_{Pub,UID} = f_0(r_{UID})f_0(\mu r_{Pub,UID}) = f_0(r_{UID} + \mu r_{Pub,UID})$, $Y_{Pub} = f_0(r_{Pub})$, and $Z_{Pub,UID} = f_2(r_{UID}I_{Pub})$, here $r_{Pub,UID} \in \mathbb{Z}^*_{n'}$ is a random number chosen by the simulator. When the adversary requests for a new attribute A_i that has not been used before, the simulator randomly chooses $I_i, k_i, h_i \in \mathbb{Z}^*_{n'}$ and $S_i = f_0(h_i) \in G_s$, $T_i = f_1(h_i) \in G_t$ to simulates the process for setting up an attribute authority. For each attribute key request made from the adversary, the simulator computes $X_{i,UID} = \varphi^{r_{UID}} S_i^{r_i} = f_0(r_{UID} + h_i r_i)$, $Y_i = \varphi^{r_i} = f_0(r_i)$, and $Z_{i,UID} = e(\varphi, \psi)^{r_{UID}I_i} = f_2(r_{UID}I_i)$, where r_i is a random number chosen from $\mathbb{Z}^*_{n'}$. The simulator passes all these values to the adversary as the attribute keys associated with A_i.
- *Challenge.* When the adversary asks for a challenge, the simulator flips a coin b and chooses a random value $s \in \mathbb{Z}^*_{n'}$. If $b = 1$, the simulator calculates $C = f_2(\alpha s)$; if $b = 0$, it picks a random value $s' \in \mathbb{Z}^*_{n'}$ and calculates $C = f_2(s')$. In addition, it calculates $C' = \varphi^{\beta s}$ and $C'' = Enc_K(ROOT)$. It also computes other components of the ciphertext following *Encrypt*: $C_{1,n} = f_1((I_{n-1} - I_n)l_n)$, $C_{2,n} = f_1(h_n(I_{n-1} - I_n)l_n)$, and $C_{3,n} = f_3((k_n t_n)^{-1})$, where $h_n \in \mathbb{Z}^*_{n'}$ is a random number chosen by the simulator.
- *Phase 2.* The simulator interacts with the adversary similar as in *Phase 1* with the exception that the adversary could not acquire attribute keys enabling a single user to satisfy the access policy $\mathbb{A}$. The output of this step is similar to that of Phase 1 except that the simulator acquires more user IDs and attributes in this step.

From the above game, we can see that the adversary only acquires the string representation of random values in $\mathbb{Z}^*_{n'}$, $\mathbb{Z}_n$ and combinations of the values. We can model all the queries as rational functions and further assume that different terms always result in different string representations [12]. As shown in [12], the probability that two terms share the same string representation is $O(q^2/n)$, where q is the number of queries made by the adversary. We assume in the rest of the proof that no such collision happens.

Now we argue that the adversary's views are identically distributed between the two cases when $C = f_1(\alpha s)(b = 1)$ and when $C = f_1(s')(b = 0)$. As a matter of fact, what the adversary can view from the modified game with the simulator are independent elements that are uniformly chosen and the only operation that the adversary can do on these elements is to test if two of them are equal or not. Thus, the situation that the views of the adversary differ can only happen when there are two different terms ν_1 and ν_2 that are equal when $b = 1$. Since αs and s' only occur in group G_1, the results from f_1 cannot be paired. Queries by the adversary can only be in the form of additive terms. Then we have: $\nu_1 - \nu_2 = \gamma\alpha s - \gamma' s'$, where γ is a constant. By transformation, we have $\nu_1 - \nu_2 + \gamma' s' = \gamma\alpha s$. This implies that by deliberately constructing a query $\nu_1 - \nu_2 + \gamma' s'$, the adversary may be able to get the value of $e(g, g)^{\gamma\alpha s}$. Now we prove that such a query cannot be constructed by the adversary based on the information it gets from the game.

In fact, the information that an adversary can acquire from this game can be listed as in Table 4. This table excludes values related to L_{UID} as it has nothing related to αs. To construct the desired value, the adversary can map two elements from G_s and G_t into one element in G_1. He can also use elements in Z_n to change the exponentials. From this table, we can easily see that to obtain a value containing αs, the adversary can pair βs and $(\alpha + r_{UID})/\beta$ to get $\alpha s + r_{UID}s$ in G_1. In fact, this is the only way to get a term containing αs. But it is not feasible in that both βs and $(\alpha + r_{UID})/\beta$ belong to G_t while the pairing requires one element from G_s and G_t each.

Therefore, based on the information an adversary can get from the proposed scheme, the attacker can not differentiate a random ciphertext from an authentic one. The security of the proposed scheme is proved.

ACKNOWLEDGMENTS

This research is supported by ONR Young Investigator Program(award NO. N00014-10-1-0714), Naval Research Lab (award NO. N00173-15-G017), ARO (award NO. W911NF-11-1-0191), NSF SaTC CNS (award NO. 1528099), the NATO Science for Peace & Security Multi-Year Project (MD.SFPP 984425), the National 973 Program (Grant No. 2013CB329601), and the National Natural Science Foundation of China (Grant NO. 61472032).

REFERENCES

[1] Cisco, "Cisco visual networking index: Forecast and methodology, 2014-2019," (2015) [online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html

[2] A. Carzaniga, M. Rutherford, and A. Wolf, "A routing scheme for content-based networking," in *Proc. 23rd Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2004, pp. 918–928.

[3] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *Proc. Conf. Appl. Technol. Archit. Protocols Comput. Commun.*, 2007, pp. 181–192.

[4] C. Dannewitz, J. Golic, B. Ohlman, and B. Ahlgren, "Secure naming for a network of information," in *Proc. INFOCOM IEEE Conf. Comput. Commun. Workshops*, 2010, pp. 1–6.

[5] N. Fotiou, P. Nikander, D. Trossen, and G. Polyzos, "Developing information networking further: From PSIRP to PURSUIT," in *Proc. 7th Int. ICST Conf. Broadband Commun. Netw. Syst.*, 2012, pp. 1–13.

[6] Named data networking (2015) [Online]. Available: http://named-data.net

[7] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proc. Sigcomm Workshop Inf.-Centric Netw.*, 2012, pp. 55–60.

[8] Y. Sun, S. K. Fayaz, Y. Guo, V. Sekar, Y. Jin, M. A. Kaafar, and S. Uhlig, "Trace-driven analysis of ICN caching algorithms on video-on-demand workloads," in *Proc. 10th ACM Int. Conf. Emerging Netw. Experiments Technol.*, 2014, pp. 363–376.

[9] Y. Zhu, G. J. Ahn, H. Hu, D. Ma, and S. Wang, "Role-based cryptosystem: A new cryptographic RBAC system based on role-key hierarchy," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 12, pp. 2138–2153, Dec. 2013.

[10] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Access control enforcement delegation for information-centric networking architectures," in *Proc. 2nd Edition ICN Workshop Inf.-Centric Net.*, 2012, pp. 85–90.

[11] S. Singh, "A trust based approach for secure access control in information centric network," *Int. J. Inf. Net. Security*, vol. 1, pp. 97–104, 2012.

[12] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Security Privacy*, 2007, pp. 321–334.

[13] S. Yu, K. Ren, and W. Lou, "Attribute-based on-demand multicast group setup with membership anonymity," in *Proc. 4th Int. Conf. Security Privacy Commun. Netow.*, 2008, pp. 18:1–18:6.

[14] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. 30th Annu. Int. Conf. Theory Appl. Cryptographic Techniq. Adv. Cryptology*, 2011, pp. 568–588.

[15] S. Arianfar, T. Koponen, B. Raghavan, and S. Shenker, "On preserving privacy in content-oriented networks," in *Proc. ACM SIGCOMM Workshop Inf.-Centric Netw.*, 2011, pp. 19–24.

[16] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Proc. 6th Int. Conf. Appl. Cryptography Netw. Security*, 2008, pp. 111–129.

[17] D. Huang, Z. Zhou, and Z. Yan, "Gradual identity exposure using attribute-based encryption," in *Proc. IEEE 2nd Int. Conf. Social Comput.*, 2010, pp. 881–888.

[18] Y. Zhu, D. Huang, C. J. Hu, and X. Wang, "From RBAC to ABAC: Constructing flexible data access control for cloud storage services," *IEEE Trans. Serv. Comput.*, vol. 8, no. 4, pp. 601–616, Jul./Aug. 2015.

[19] Y. Zhu, H. Hu, G.-J. Ahn, M. Yu, and H. Zhao, "Comparison-based encryption for fine-grained access control in clouds," in *Proc. 2nd ACM Conf. Data Appl. Security Privacy*, 2012, pp. 105–116.

[20] Z. Wang, D. Huang, Y. Zhu, B. Li, and C.-J. Chung, "Efficient attribute-based comparable data access control," *IEEE Trans. Comput.*, vol. 64, no. 12, pp. 3430–3443, Dec. 2015.

[21] M. Chase, "Multi-authority attribute based encryption," in *Proc. 4th Conf. Theory Cryptography*, 2007, pp. 515–534.

[22] M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. 16th ACM Conf. Comput. Commun. Security*, 2009, pp. 121–130.

[23] Z. Liu, Z. Cao, Q. Huang, D. S. Wong, and T. H. Yuen, "Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles," in *Proc. 16th Eur. Conf. Res. Comput. Security*, 2011, pp. 278–297.

[24] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein, "OWL web ontology language reference," (2004) [Onlilne]. Available: https://www.w3.org/TR/2004/REC-owl-ref-20040210/

[25] SPARQL query language for RDF (2008) [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[26] B. Li, Z. Wang, and D. Huang, "An efficient and anonymous attribute-based group setup scheme," in *Proc. IEEE Global Telecommun. Conf.*, 2013, pp. 861–866.

[27] R. Biswas, K. Chowdhury, and D. Agrawal, "Attribute allocation and retrieval scheme for large-scale sensor networks," *Int. J. Wireless Inf. Netw.*, vol. 13, no. 4, pp. 303–315, 2006.

[28] ndn-cxx: NDN C++ library with eXperimental eXtensions 0.3.1-6-ga76bbc9 documentation (2015) [Online]. Available: http://named-data.net/doc/ndn-cxx/current/

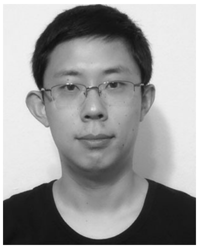
[29] NFD - Named data networking forwarding daemon 0.3.1 documentation (2015) [Online]. Available: http://named-data.net/doc/NFD/current/

[30] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. 14th ACM Conf. Comput. Commun. Security*, 2007, pp. 456–465.

[31] B. Lynn, "PBC library the pairing-based cryptography library," in *http://crypto.stanford.edu/pbc/*, Accessed March 2014.

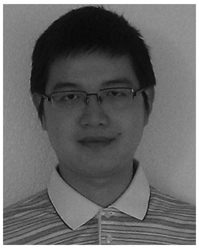
[32] P. Library. (2015). Curve Param generation [Online]. Available: https://crypto.stanford.edu/pbc/manual/ch05s01.html

[33] V. Shoup, "Lower bounds for discrete logarithms and related problems," in *Proc. Int. Conf. Theory Appl. Cryptographic Techniq.*, 1997, pp. 256–266.

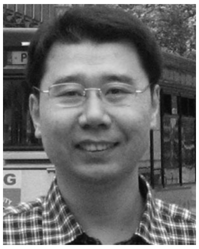


Bing Li received the BS degree from the Shandong University in 2008 and the MS degree from the Beijing University of Posts & Telecommunications in 2011. He is currently working toward the PhD degree at the Arizona State University. His research interests include ad hoc network security, cloud computing, and robot systems.

Dijiang Huang (Member 2000, Senior Member 2011) received the BS degree from the Beijing University of Posts & Telecommunications, China 1995. He received the Master of Science and PhD degrees from the University of Missouri-Kansas City in 2001 and 2004, respectively. Both majored in computer science and telecommunications. He joined ASU in 2005 in the Department of Computer Science and Engineering as an assistant professor. From 2011, he has been an associate professor in the School of Computing Informatics and Decision Systems Engineering. His current research interests include computer and network security, mobile ad hoc networks, network virtualization, and mobile cloud computing. His research is supported by Federal Agencies NSF, ONR, NRL, ARO, and NATO, and organizations such as Consortium of Embedded System (CES), Hewlett-Packard, and China Mobile. He received the ONR Young Investigator Award and HP Innovation Research Program (IRP) Award. He is a co-founder of Athena Network Solutions LLC (ATHENETS). He is currently leading the Secure Networking and Computing (SNAC) research group at ASU. He is a senior member of the IEEE.



Zhijie Wang received the BS and MS degrees from the Beijing University of Posts & Telecommunications, in 2007 and 2010, respectively. He is currently working toward the PhD degree at Arizona State University. His research interests include wireless networking, applied cryptography and cloud computing.



Yan Zhu is a professor in the School of Computer and Communication Engineering, University of Science and Technology, Beijing, China. He was an associate professor at the Peking University in China from 2007 to 2012. He was a visiting associate professor in the Arizona State University from 2008 to 2009, and a visiting research investigator of the University of Michigan-Dearborn in 2012. His research interests include cryptography, secure computation, and network security. He is a member of the IEEE.