DataClouds: Enabling Community-based Data-Centric Services over Internet of Things

Hao Yue, *Student Member, IEEE,* Linke Guo, *Student Member, IEEE,* Ruidong Li, *Member, IEEE,*
Hitoshi Asaeda, *Senior Member, IEEE,* Yuguang Fang, *Fellow, IEEE*

***Abstract*—Internet of Things (IoT) is emerging as one of the major trends for the next evolution of the Internet, where billions of physical objects or things (including but not limited to humans) will be connected over Internet and vast amount of information data will be shared among them. However, current Internet was built on a host-centric communication model, which was primarily designed for meeting the demand of pair-wise peer-to-peer communications and cannot well accommodate various advanced data-centric services boosted by IoT in which users care about content and are oblivious to locations where the content is stored. In this paper, we propose a novel architecture for future Internet based on Information-Centric Networking (ICN), which is called DataClouds, to better accommodate data-centric services. Different from existing ICN-based architectures, we take the sharing nature of data-centric services under IoT into consideration and introduce logically and physically formed communities as the basic building blocks to construct the network so that data could be more efficiently shared and disseminated among interested users. We also elaborate on several fundamental design challenges for Internet under this new architecture and show that DataClouds could offer more efficient and flexible solutions than traditional ICN-based architectures.**

***Index Terms*—Internet of Things, Information-Centric Networking, Internet architecture, Data-centric Services.**

I. Introduction

FOR several decades, Internet serves as a worldwide information infrastructure for humans to communicate with one another. Nowadays, due to the technological advances in low-cost sensors, radio frequency identification (RFID), scalable cloud computing and ubiquitous wireless connectivity, more and more physical objects and things are able to connect to Internet and interact with the physical world around, which makes the Internet evolve to a new era, known as the "Internet of Things" (IoT) [1]–[3]. Under IoT, a vast amount of data, i.e., big data, will be available and shared over Internet among interested parties. We can extract more valuable and insightful knowledge about the state of objects or surrounding environment, and then have a better understanding and control over them, which will significantly improve the quality of our life. Based on a recent white paper from Cisco, there will be 50 billions devices connected to Internet by 2020 and IoT will create 14.4 Trillion dollars of value at stake for industries in the next decade.

IoT has boosted various advanced services over the Internet and most of them are data-centric in nature, where data is collected, disseminated and shared among a certain group of objects and humans with common interests. The essential characteristic of such data-centric services is that users care about the content themselves and are oblivious to the locations where the desired data resides. Here, we take Smart Cities as an example for illustration. Smart Cities are one of the most outstanding creations of IoT [4]–[6], and data-centric services exist everywhere in different systems in Smart Cities, as shown in Fig. 1. For instance, in a healthcare system, body area networks consisting of small sensors are widely deployed to monitor patients' health status like heart rate and blood pressure. The measured data is sent to doctors for healthcare advices, and could also be accessed by the patients and their relatives. The doctors share the treatment data with the patients and concerned parties (e.g., relatives and friends), and supervisory signals could be sent back to the sensors in body area networks for triggering corresponding actions. In a transportation system, drivers can collect information about road conditions and traffic status with the RFID tags attached to their cars or roadside facilities, and share the information with other drivers to avoid traffic congestion. The police can also use the data to make a better control and dispatch over the whole system. By using IoT technologies, we can build a smart environment, such as a smart city, we are living in, for better life.

Although these data-centric services enabled under IoT are very attractive and beneficial, they cannot be well supported by the current Internet. Built on a host-centric communication model, the current Internet was designed for end-to-end communications between pairs of nodes. However, as we mentioned above, the essence of the data-centric services is information dissemination and sharing, where users focus on data and information themselves rather than on the end node to communicate with. This mismatch between the traditional network architecture and the new service model will inevitably lead to inefficient resource utilization, unnecessary communication overhead and severe network performance degradation when running such data-centric services over the current Internet. Besides, mobile users can be observed almost

This work was partially supported by the U.S. National Science Foundation under Grants CNS-1343356.

H. Yue and Y. Fang are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA e-mail: {hyue@, fang@ece.}ufl.edu.

L. Guo is currently with Department of Electrical and Computer Engineering, Binghamton University, State University of New York, email: lguo@binghamton.edu.

R. Li and H. Asaeda are with the National Institute of Information and Communication Technology 4-2-1, NuKui-Katamachi, Koganei, Tokyo, 184-8795, Japan e-mail: {lrd@, asaeda@}nict.go.jp.

everywhere in IoT from sensing devices attached to animals and cars to humans with smartphones and tablets. However, the current Internet architecture concentrates on relatively static hosts and does not take user mobility into consideration. Though some add-on mechanisms, such as mobile IP, can be used to alleviate this issue at the cost of increasing the complexity of overall network architecture, mobility still cannot be effectively supported under the current Internet [7], mainly due to the underlying host-centric communication model.

Recently, Information-Centric Networking (ICN) has emerged as a new paradigm for future Internet [8], [9]. Rather than relying on the traditional host-centric communication model, ICN takes a novel information-centric communication model, which is well-suited to content dissemination and sharing. Under this model, information contents are named and addressed regardless of their locations, and therefore could be stored anywhere in the network [10]. Each time when a user intends to access some data or services, instead of first determining the destination host for communication, it sends its interest to the network, which will locate the best source of the desired data based on its name and return it to the user. Since the information-centric communication model well matches content-dissemination-based services, ICN is a promising candidate for the architecture of future Internet under IoT.

Several ICN-based architectures have already been presented for future Internet, such as DONA [11], PURSUIT [12], [13], CCN [14]–[16], SAIL [17] and CONVERGENCE [18]. Although these architectures realize the basic functionalities of ICN, none of them are primarily designed for enabling data-centric services under IoT, and therefore they suffer from some common drawbacks. First, in IoT where billions of objects are connected and communicate over the Internet, a huge number of users may intend to access their interested data at the same time. Under the existing ICN-based architectures, users do not hold any reference to data publishers. Hence, the network has to search for content source and construct delivery routes for each data access request, which will lead to a significant increase in network traffic, data access delay, and control overhead for the data-centric services. Second, in ICN, the locations of data publishers must be continuously tracked for name resolution (i.e., correctly map a data name to the location where it stores). The above ICN-based architectures lack effective hierarchical structures for such tracking and the control messages have to be disseminated through the whole network, which results in a vast amount of control overhead for mobility management. Third, without centralized control in users of data-centric services, the caching strategies used in the existing ICN-based architectures work in a distributed manner and are heuristic in nature, which are not able to ensure optimality of caching performance and also cause inefficiency in resource utilization. Moreover, ICN method can create severe privacy threats to users of some data-centric services in IoT, like healthcare, since users reveal their interest to the network and the name of data is available to all the nodes forwarding it. The previously proposed architectures [9], [19], [20] concentrate on information confidentiality and integrity, where the privacy issue has not been well studied and addressed.

A community-oriented ICN network architecture CORIN was proposed in [21]. By considering the aforementioned observations on the characteristics of data-centric services under IoT and the drawbacks of existing ICN-based architectures, we extend CORIN and design a novel architecture for the future Internet, called *DataClouds*, to better accommodate IoT and enhance users' experience in data dissemination and information sharing. The basic building blocks of DataClouds are called communities, which consist of users with common interests in data and information. We carefully construct and maintain the structures of communities and ensure that the users within the same community are well connected in the network so that desired data can be efficiently collected, disseminated and shared among them. With the introduction of communities, the whole Internet can be abstractedly viewed as a collection of cloudlets of data, where each cloudlet corresponds to a community. A cloudlet carries the data which the users in a community are interested in, and can disseminate the data to the users based on their requests. Moreover, a cloudlet can float as the users move around in the network, shrink or expand as new users join or current members leave the community, and split into several cloudlets or merge with other cloudlets based on users' different needs. The data management and topology formation are just like the behaviors of floating clouds, which motivates us to use the name DataClouds for this novel architecture.

As an important note, we want to point out that the DataClouds architecture can be easily integrated with our recently proposed cognitive capacity harvesting (CCH) network architecture to harvest wireless network resource to transmit tremendous data around for billions of objects over IoT [22]–[24]. In CCH, cognitive radio routers can be deployed together with existing access points (i.e., access points in WLANs or base stations in cellular systems) to form spectrum clouds to harvest outband spectral resource to forward data closer to the end users in order to mitigate congestion on inband spectrum or shared spectrum. As most of data traffic over IoT is relatively delay-tolerant, CCH provides the ideal resource harvesting facilities to deal with ever increasing data contents over IoT. Detailed integration will be investigated in the future.

The rest of this paper is organized as follows. The related work is reviewed in Section II. In Section III, we introduce the system architecture of DataClouds and discuss its attractive features for data-centric services over IoT. Mobility management, in-networking caching, and security and privacy issues under DataClouds are studied in Section IV, V, and VI, respectively. Finally, we draw the concluding remarks in Section VII.

II. Related Work

Several ICN-based Internet architectures have been proposed in the past few years. In the Data Oriented Network Architecture (DONA), content items are associated with flat names. A collection of Resolution Handlers (RHs) are introduced into the network for name resolution. In order to advertise a data item, the owner registers the data name with the local RH, which will distribute it to other RHs in a

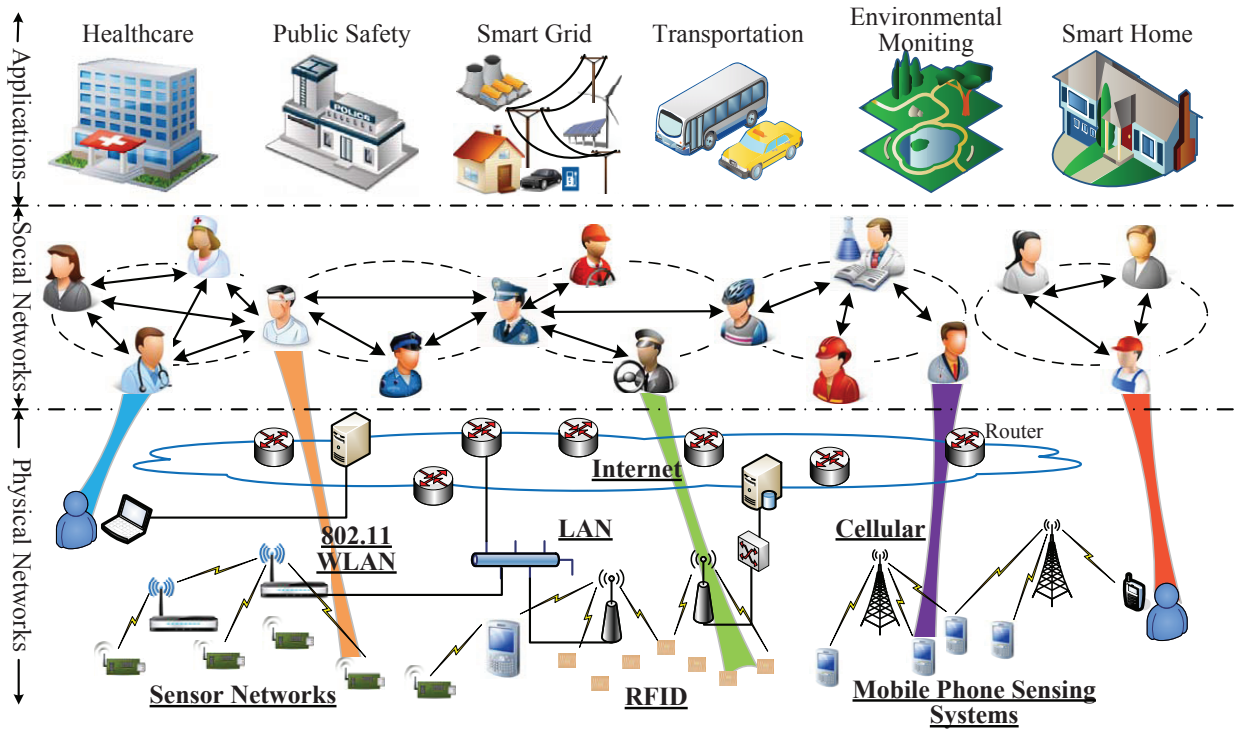

Fig. 1. Smart Cities over IoT.

hierarchical fashion. When a user is interested in the data and intends to retrieve it from the network, it sends a request to the RHs, which find the corresponding owner based on the record and forward the request to it. Then, the desired data will be transmitted from the owner to the interested user. Different from DONA, the Content Centric Networking (CCN) uses hierarchical names for data items. The Content Routers (CRs) maintain the location information for data items, and forward request messages and data items between the owner and the interested users. Each CR also serves as a local cache for data items that have been transmitted through it. In the architecture designed in the Publish Subscribe Internet Technology (PURSUIT) project, data is identified with two IDs, called the scope ID and the rendezvous ID. PURSUIT consists of three modules: rendezvous, topology management, and forwarding. Name resolution is achieved by the Rendezvous Network (RENE) in the rendezvous module, which searches for the location of the desired data and instructs the topology management module to create a route for data transmission. Finally, the data is delivered via a set of Forwarding Nodes (FNs) in the forwarding module to the destination. As a combination of existing design, the architecture in the Network of Information (NetInf) offers two models for retrieving named data items, via name resolution as in the PURSUIT and via name-based routing as in the CCN. The names used in NetInf also have the characteristics of both flat names and hierarchical names.

In contrast to these existing architectures, in DataClouds, we insert a new hierarchical level, referred to as communities, into the network architecture. A community includes the users with common interests in data and the whole network consists of various communities constructed according to different data-centric services of interest. Data dissemination is achieved via community-oriented communications, where name resolution and data routing are restricted within each community. Dividing users of data-centric services in IoT into a collection of communities could also enhance management over the whole network, which facilitates the design of better mechanisms to support user mobility, achieve in-network caching, and protect user privacy, as we will show later in this paper.

III. System Architecture

A. Overview

In [21], we have introduced the concept of "community-oriented communications" for the first time and proposed a community-oriented route coordination (CORIN) system. Under "community-oriented communications", users with shared interests are grouped into different communities, and communications occur among the users within the same community. It has been shown that efficient information dissemination can be achieved under CORIN with low control overhead. The new architecture DataClouds is extended from CORIN. Fig. 2 shows the overview on the network architecture of DataClouds, which involves two different spaces: the *logical space* and the *physical space*. The logical space captures user relationships and interests in data content. Here, we connect the users that share common interests in one data-centric service, which will divide the users into different groups, called *logical communities*. The physical space manages physical network connections among users. We introduce three new kinds of entities: Forwarding Nodes (FNs), Forwarding and Caching Nodes (FCNs), and Community Rendezvous Points

(CRPs), which will be elaborated later. With these entities, the users with same interests in a logical community are well connected and form a physical community in the physical space, as shown in Fig. 2.

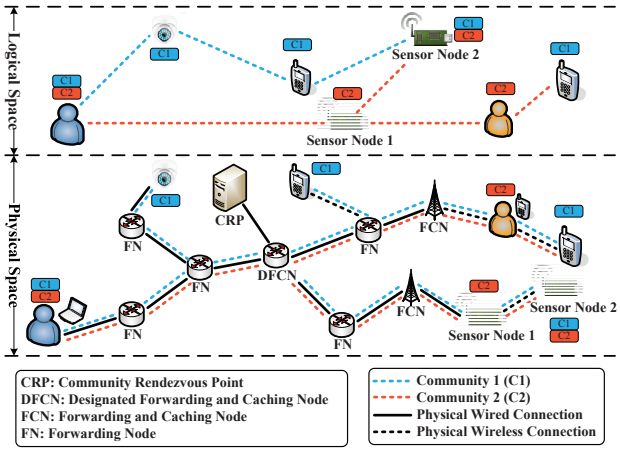

Fig. 2. The system architecture of DataClouds.

B. Naming

Similar to other ICN-based architectures, in DataClouds, a data item is associated with an identifier, called *information identifier (IID)*. In addition, we also have a name for every community, which is referred to as *community identifier (CID)*. For example, there are two logical communities with CIDs C1 and C2 in the logical space as illustrated in Fig. 2. The structure of the names associated with data items is one of the fundamental design issues for ICN-based architectures. Several methods have been used in the literature, where the names can be flat or hierarchical, human-readable or not, as we mentioned in Section II. Here, we introduce one candidate of the naming strategies for DataClouds. As done in [21], the format of the CIDs could be defined as `<Service Type|Hierarchical Community Name (HCN)|Domain Name>`. Taking the Smart Cities shown in Fig. 1 as an example, we might have a community with CID as `<Healthcare|Florida|Health Shands Hospital|Bob|Domain 1>`. Here, `Healthcare` is the `Service Type`. `Florida|Health Shands Hospital|Bob` is the `HCN`, which describes the community under this service. `Domain 1` indicates the administrative domain where the community is created. Since one data item in DataClouds is shared among the interested users within a community, the IIDs could have the format as `<Service Type|HCN|Target|Domain Name>`. For example, the IID of the data items disseminated within the above community can be `<Healthcare|Florida|Health Shands Hospital|Bob|Heart Rate|Domain 1>`. With the CIDs and IIDs, a user in DataClouds can easily find other users with the same interest and access the desired data. Note that one user can be a member of multiple communities based on her interests.

C. Network Entities

DataClouds consists of several kinds of new network entities, which are described as follows:

- **Community Rendezvous Points (CRPs):** In DataClouds, there is one CRP in each administrative domain, which records the CIDs of all the communities in this domain. The main functionalities of CRPs include searching for desired communities based on users' interests and assisting them to join the communities in physical space.
- **Forwarding Nodes (FNs):** FNs work as intermediate nodes to forward data and control messages. The routing function at FNs is achieved with two modules, called the *Forwarding Information Base (FIB)* and the *Request Pending Table (RPT)*. FIB maintains the CIDs of different communities and the corresponding incoming and outgoing interfaces for data forwarding. The RPT records the routes created between users and the CRP for the transmissions of control messages when the users initiate new communities or join existing communities.
- **Forwarding and Caching Nodes (FCNs):** FCNs have all the functionalities of FNs. In addition, FCNs are equipped with memories to cache/store data items in order to facilitate future data retrieval.
- **Designated Forwarding and Caching Nodes (DFCNs):** When a new community is initiated, the CRP will designate one FCN to it, which serves as the root of the communication structure of the community in physical space, i.e., every member in this community must be connected to this FCN so that it can share data with or retrieve data from others.

D. Community Management

In DataClouds, communities are created and maintained through the following operations:

- **Community Initialization:** When a user has data to disseminate but there are no communities existing for them, the user can create a new community for the users with the same interest to share the data. To do so, the user needs to register the new community at the local CRP, and the CRP will assign a DFCN to the community. After that, a route is established between the user and the DFCN, which forms the original communication structure of the community.
- **Community Joining:** After querying the CRP, a user can discover and join the communities in the network that meet her interest in order to publish or receive the desired data. Similar to community initialization, the user has to find a route and connects herself to the DFCNs of the corresponding communities. In this way, the community, or the cloudlet is expanding.
- **Community Leaving:** Since the interests of users might change over time, they can choose not to share data with other users in certain community any more by leaving it. When a user decides to leave a community, it needs to inform the DFCN so that the route between the user and the community can be removed from the communication

structure. In this way, the community or the cloudlet is shrinking.

- **Community Deletion:** When there are no users in a community, the DFCN will inform the CRP to delete the corresponding record for this community. The DFCN is freed from that community and can be reassigned to new communities in the future. Thus, the community or cloudlet disappears.

E. Merits

DataClouds serves as a viable approach to enabling data-centric services over IoT, which could address almost all issues we have discussed about the existing ICN-based architectures. First, in DataClouds, since the users interested in the same service are well connected via the communication structures of communities, data dissemination could be efficiently achieved among them without unnecessary duplicate transmissions, which significantly reduces the communication overhead and improves the resource utilization. Second, after a user joins a new community, it can continuously access the desired data via the communication structure of the community without finding the data source and establishing a route every time, which will drastically ease the workload for name resolution and reduce the control overhead in the network. More importantly, the user only manages to connect to the edge of the community, just as done for a mobile user in cellular systems. In this way, the community-based DataClouds provides a natural hierarchical network organization, which can effectively address the scalability issue. Third, in DataClouds, different interests can be restricted in individual communities. Hence, users can choose the communities to join based on their own interests and the tussles among users with different or controversial interests can be delimited. Fourth, better in-network caching strategies could be performed with the information on the communication structures of different communities to facilitate data dissemination and minimize the transmission cost. Finally, in DataClouds, when a user moves to a new location, it only needs to locally reconstruct the connections to its communities, which could reduce the control signaling overhead due to mobility management.

In the subsequent development, we will discuss several fundamental issues on network architectural design under DataClouds and illustrate the merits we have mentioned above in details.

IV. Mobility Management

In traditional ICN-based architectures, the network has to track the movement of data publishers for name resolution and route construction. In DataClouds, users receive data-centric services from the network via communication structures of communities. When a user moves around within the network, we need to update its connections to the communities for continuous and successful data access and sharing.

A. Mobility Models

We will use the following terminologies in the rest of this section. Given a community, the *home-domain* is referred to as the administrative domain where the community is created, and the other administrative domains in the network are called the *foreign-domains*, relative to the home domain. The home CRP of a community is defined as the CRP serving its home domain. We also have the home DFCN, which is the DFCN assigned to the community by the home CRP during community initiation. A FN/FCN is called an edge FN/FCN if it directly connects to end users.

Based on the domains where a user roams around, user movement in DataClouds can be classified into two broad categories: home-domain roaming and foreign-domain roaming. For either of these two cases, we are interested in two different mobility models: *user-mobility* and *community-mobility*. In user-mobility, FN/FCNs are static and only users move around in the system. In this case, mobile users need to maintain their connections to their communities in order to continuously publish data to or receive data from them. In contrast, in community-mobility, FN/FCNs move together with users as a whole, e.g., a body area network or a railway network. Different from user-mobility, in this case, the communication structure of the community remains unchanged. The home DFCN must inform the local CRP about the movement of the community so that the CRP is aware of its new connection and can direct new users to join in the future. Obviously, a community can be expandable or condensible according to the movement of the community members, i.e., new FN/FCNs are added into the community for making new connections with the community members or the FN/FCNs along the old connections are removed from the community as the members move. Note that here we will not consider the case where the FNs/FCNs move within the network independently of the users due to lack of practical applications. In the following subsections, we elaborate the design of some basic mobility management mechanisms for DataClouds.

B. Home-Domain Mobility Management

1) User-Mobility: Each time when a user accesses the data in a community, it will first communicate with an edge FN/FCN and identify the CRP which serves the current domain. If the CRP is its home CRP and the edge FN/FCN remains the same as before, there is no movement for the user between two consecutive communications. If the user finds that it is still in the home domain, but the edge FN/FCN changes, it will perform the following steps for home-domain mobility management:

1. The user sends a CONNECTION UPDATE message with the CID towards the home DFCN.
2. When an intermediate FN/FCN receives the CONNECTION UPDATE message, it examines its own FIB table. If the FN/FCN has no record for the corresponding community in the table, it will create one entry for this community and add the incoming interface into it for future routing. Then, it forwards the message towards the DFCN.
3. If there already exists one matching record in the FIB table of a FN/FCN, which indicates that the FN/FCN is in the communication structure, it will add the new

incoming interface into the record so that a new route between the user and the community will be added to the communication structure. After that, the FN/FCN discards the CONNECTION UPDATE message without further forwarding.

4. When an edge FN/FCN detects the absence of the user, it will delete the interface assigned to the user from its record in the FIB table. If there is no user connected to the community via this edge FN/FCN, it will also send a PATH DELETE message towards the upstream nodes in order to remove the old path for the user from the community.

2) Community-Mobility: When an entire community moves within its home domain, the home DFCN must periodically update its connection to the CRP so that new users can join the community and establish a connection to the DFCN. In this scenario, the local CRP will record a valid path towards each DFCN in the domain. When one community moves to a new place, the home DFCN will send a CONNECTION UPDATE message to the local CRP, which is similar to the transmission of CONNECTION UPDATE message in the user-mobility case. When neighboring FN/FCNs at the old place discover that the DFCN has moved out of this area, they will generate PATH DELETE message and transmit it along the old path towards the local CRP. When intermediate FN/FCNs receive this message, they will remove the old path from the system.

C. Foreign-Domain Mobility Management

1) User-Mobility: A mobile user determines the foreign-domain roaming has occurred when it discovers that it is in a foreign domain with a different edge FN/FCN. Then, the following steps will be carried out:

1. The user sends a CONNECTION UPDATE message with the CID to the local CRP.
2. Since the local CRP serves a foreign domain and might have no record for this community, it will forward the message to the CRP at the upper level for further search until a matching record is found. Then, the CRP at the upper level transmits the CONNECTION UPDATE message towards the home CRP of the community. At the same time, each intermediate CRP also designates one DFCN as an intermediate node for route establishment. Finally, when the CONNECTION UPDATE message arrives at the home DFCN, a new path consisting of FNs, FCNs, and DFCNs will be created between the foreign-domain and the home-domain of the community.
3. The old edge FN/FCN sends the PATH DELETE message to remove the old path for the user, which is similar to that in the home-domain mobility management.

2) Community-Mobility: When a mobile DFCN observes that the local CRP in the current domain is different from the home CRP, it will perform connection update for foreign-domain mobility management. The CONNECTION UPDATE message sent from the DFCN will first arrive at the local CRP of the current domain, where the community will be registered. Then, the CONNECTION UPDATE message is forwarded and might traverse through multiple CRPs at different levels to reach the CRP serving the last domain where the community resides. When the CRP receives the message, it will deregister the community and send a PATH DELETE message to remove the old path.

D. Discussion

We have described some basic mobility management mechanisms for DataClouds. The basic idea here is to maintain the connections between the users and their communities and the connections between the communities and the whole network when the locations of users change. We can observe that the community structures employed in DataClouds drastically facilitate user mobility management. On one hand, since user-mobility is much more common in practice as compared to community-mobility, most of control signal is exchanged within each community in DataClouds, which will significantly reduce the communication overhead for mobility management. On the other hand, in sharing-based data-centric services, a user is potentially both the data sender and receiver. The mobility management mechanisms in DataClouds treat data senders and receivers in a unified manner, and reduce the latency for locating the data source by keeping it well connected to all the interested receivers.

Mobility management has been well studied under cellular systems, and we can borrow the ideas from some existing techniques there, such as pointer forwarding [25]–[29] and local anchoring [25]–[27], [30], to further improve the performance of the basic mobility management mechanisms. For example, when a user moves into a foreign domain that is far away from its home domain, the CRP of the current domain could assign a local FCN as its anchor. Then, when the user moves around within this domain, it could update its new connection with the anchor rather than contacting the remote home CRP. In addition, we can take the mobility pattern of individual users into consideration to design more intelligent mobility management mechanisms. For example, if a user constantly moves between one edge FN/FCN in domain A and another edge FN/FCN in domain B (e.g., a doctor moves between the hospital and her home), the mobility management mechanisms could maintain two routes in the communication structure of every community the user has joined, which are from the two edge FN/FCNs to its home DFCN, to effectively reduce the control overhead for connection update [31].

V. Caching

In-network caching is a fundamental feature of ICN-based architectures. In DataClouds, data and information shared within communities can be distributively cached at a collection of FCNs. Later, when a user intends to access the data, it can retrieve them from the closest FCNs with short delay and low signaling traffic.

In the current literature, most of the research on data caching carried out under traditional ICN-based architectures [11], [12], [14], [32] focuses on caching-aware name resolution and routing mechanism design. Existing caching strategies on selection of the caching locations for different data items

are mostly heuristic in nature, and therefore cannot ensure optimality due to lack of centralized management for users. Since we carefully maintain the communication structures of communities in DataClouds, we are able to design more advanced caching strategies to maximize the efficiency of data access for users of data-centric services under IoT.

A. Basic Idea

Consider a domain in DataClouds, which contains one CRP, one DFCN and a number of FNs. We start with a simplified version of the caching problem and concentrate on one community. Let $\mathcal{U}$ denote the set of all the users in the community. We assume that users are static and the number of users does not change over time, i.e., there is no community joining and leaving. Each user is associated with an edge FN/FCN to connect to the network. Let $\mathcal{N}$ denote the set of all the FNs installed in the domain. Our goal is to select a subset of FNs from $\mathcal{N}$ and cache data at them (i.e., replace them with FCNs) so that the data can be retrieved by the users in the community with minimum cost. For illustrative purpose, in the rest of this section, we consider the delay for data retrieval as the cost function. However, the formulations derived here can be easily extended to other cost functions, such as signaling traffic or energy consumption.

Let $\mathcal{D}$ denote the set of all the data items that need to be cached for the community. Given the transmission rates of the links among the FNs and DFCN as well as the sizes of data items in $\mathcal{D}$, let $t_{ij}^d(s)$ stand for the time taken for user $i \in \mathcal{U}$ to retrieve the cached data $d \in \mathcal{D}$ from FN $j \in \mathcal{N}$ when the caching strategy s is used. Here, a caching strategy s will specify the set of locations where the FCNs are installed in the domain and the subset of data items stored at each location. We set $t_{ij}^d(s) = +\infty$ if data item d is not cached at FN j under strategy s. Then, the time taken for user i to receive data item d from the community, denoted as $t_i^d(s)$, could be expressed as $t_i^d(s) = \min_{j\in\mathcal{N}} t_{ij}^d(s)$. We use $t_i(s)$ to denote the delay for user i to retrieve all data items from the community under strategy s, which can be calculated as $t_i(s) = \max_{d\in\mathcal{D}} t_i^d(s)$ if we consider the delay for a user to retrieve all data items as the maximum value of the time taken for the user to retrieve each one of them. Let $\mathcal{S}$ denote the set of all the candidate strategies for caching data at the FCNs with consideration of detailed constraints such as the maximum number of FCNs deployed, the buffer size of FCNs, *etc*. Our objective is to minimize the maximum time taken for the users to access data from the community. Then, we can formulate a cost minimization problem to derive the best caching strategies, which is shown as follows:

$$\min_{s\in\mathcal{S}} \max_{i\in\mathcal{U}} (t_i(s)) \ \textbf{such that} \ t_i(s) = \max_{d\in\mathcal{D}} t_i^d(s) = \max_{d\in\mathcal{D}} \min_{j\in\mathcal{N}} t_{ij}^d(s) \tag{1}$$

B. Discussion

The above optimization problem could be extended with the consideration of more important factors to better solve the caching problem. For example, we have not considered the popularity of different data items in (1). In order to address this issue and make the optimization formulations more practical, we can have a weight $\omega_d \in (0, 1]$ associated with each data item $d \in \mathcal{D}$ to indicate its popularity. Then, the constraint in (1) will become

$$t_i(s) = \max_{d\in\mathcal{D}} \omega_d \cdot t_i^d(s) = \max_{d\in\mathcal{D}} \min_{j\in\mathcal{N}} \omega_d \cdot t_{ij}^d(s).$$

which describes the weighted data access delay for each user. Furthermore, the formulations in (1) have not taken the uncertainty in user location into account. Considering user mobility as well as community joining and leaving, the location of a user in a community may change over time. In this case, the time for user i to retrieve the cached data d at FCN j under the selection strategy s, i.e., $t_{ij}^d(s)$, will be a random variable, and (1) becomes a stochastic optimization problem. Also note that until now we focus on the caching issue for one single community. In practice, there are usually a number of communities sharing FCNs in a domain, and a user could be the member of multiple communities. One direct extension of the above formulations is to find the optimal caching strategies which minimize the overall cost for the users to retrieve data from different coexisting communities.

VI. Security and Privacy

DataClouds provides a viable platform for data dissemination in data-centric services of IoT. Unfortunately, without additional precaution, this prominent approach exposes data contents to FCNs and FNs, which may leak the detailed information about the contents and users' identities. Even worse, active adversaries may be able to collect cached items in FCNs to infer possible information about users' location, interests, social relationships, etc. Current security strategies mostly rely on end-to-end solutions, and hence will not be effective. Since DataClouds is based on community of interests, which can be characterized with attributes, the attribute-based cryptographic techniques can be used to address the security and privacy issues in DataClouds [33], [34]. In this section, by leveraging the attribute-based cryptography, we propose a built-in secure, robust, and efficient content caching and dissemination framework over DataClouds and discuss some design issues with possible solutions.

A. Trust Model

We first elaborate on the trust model. We assume that the CRPs are fully-trusted, which are responsible for security setup, parameter distribution, and managing registration and revocation of users when they join and leave a community. The FNs and FCNs are semi-trusted (i.e., curious but not malicious). The DFCNs are also semi-trusted and they help users search and locate particular named-data, in which cached encrypted searchable tables are stored and the location of a desired cache is determined.

B. Content Privacy

Content privacy is a critical issue for some data-centric services such as healthcare in DataClouds, where content

publishers cache their named-contents in FCNs and allow interested users to search and retrieve. Current ICN approaches have not addressed well on security and privacy, and hence cannot provide confidentiality protection to cached contents in FCNs. Even worse, the property of named-contents will bring in privacy breaches not only to the content itself, but also to users who are interested in this content. Therefore, we need to develop privacy-preserving mechanisms to protect content privacy in our proposed architecture.

One of the possible ways to provide content privacy is broadcast encryption [35], which allows a broadcaster to send an encrypted message to a set of receivers, each of which has a different private key. Content subscribers would be able to find entries to cached data and further decrypt it. Proxy re-encryption [36] has been discussed as a way to provide content privacy in ICN. It allows a third-party (called proxy) to re-encrypt a ciphertext which has been encrypted for a user, say, Alice, so that it can be decrypted by another user, say, Bob. Arianfar et al. in [37] describe an algorithm to mix legitimate contents with so-called "cover files". The content publisher selects a cover content to mix with a legitimate content. All of the above works have not been implemented in the real system either due to the large communications overhead or the computational overhead.

Based on our community model, we consider each community is attribute-driven (as social groups in Online Social Networks), where users join the community because they can obtain the contents that they are interested in. We categorize the named-contents into different attributes att, and content subscribers can search for the contents via verifiable attributes. For the purpose of content privacy, content publishers encrypt given content m by using the attributes of this content, e.g., health, sports, business, or transportation. Then, content subscribers can search and retrieve their interested contents only if they have the corresponding attribute-based private keys sk.

Our basic approach is described as follows [38].

1. Given the security parameter ξ, the CRP generates a parameter tuple $(p, g, G_1, G_2, e) \leftarrow \mathcal{G}(1^{\xi})$, and further outputs the public key $PK := (g, h = g^{\beta}, f = g^{1/\beta}, e(g,g)^{\alpha})$.
2. For the user with the attribute set S, the CRP outputs the corresponding private key as $sk := (D = g^{(\alpha+r)/\beta}, \forall j \in S, D_j = g^r \cdot H(j)^{r_j}, D'_j = g^{r_j})$.
3. To encrypt a content m, the content publisher chooses an access structure $\mathbb{A}$, and the ciphertext is $CT := (\mathbb{A}, \tilde{C} = m \cdot e(g,g)^{\alpha s}, C = h^s, \forall y \in \mathbb{A}, C_y = g^{q_y(0)}, C'_y = H(att(y))^{q_y(0)})$, where $q_R(0) = s$ is the secret as the root node in $\mathbb{A}$ and q_y is defined as a polynomial.
4. For users with the corresponding private keys, they can use sk to decrypt the content by computing $e(D_i, C_x)/e(D'_i, C'_x) = e(g,g)^{rq_x(0)}$ and further derive $e(g,g)^{rq_R(0)}$. The content m can be derived by $\tilde{C}/(e(C,D)/e(g,g)^{rs})$.

We show the basic approach by directly applying the Ciphertext-policy Attribute-based Encryption. Users can encrypt their contents and cache at FCNs, in order to allow content subscribers to retrieve. However, as the same reason as several other cryptographic approaches, this approach incurs intensive computational overhead on the end user side, especially for the pairing operation $e(g,g)$ on the mobile users with devices of low computational capability. Thus, advanced techniques need to be considered to reduce the computational complexity in our proposed attribute-based encryption schemes.

C. Content Verification

To guarantee the integrity and authenticity of the desired contents, we propose to design a secure and efficient signature scheme for content sharing over DataClouds. Considering an illustrative example in a mobile health system, a patient, say, Alice, shares her health data to different users, including family members, doctors, friends, and/or nearby hospitals. One of the critical issues for the content sharing between Alice and doctors is to ensure the authenticity and non-repudiation of the content. Without this assurance, doctors and nearby hospitals may not be able to use her data for diagnostic and healthcare treatment. In addition to this, the privacy of the verification process is also important for patients, where they have to reveal some of their public information for the verification. Therefore, we need to develop a privacy-preserving verification scheme to secure the content sharing over DataClouds.

Potential solutions include applications of group signature, undeniable signature, and ring signature in content verification. Group signatures [39] allow the signer to hide in a set of potential signers, and thus provide signer-ambiguity. Although the group signature provides the identity privacy for a patient in a particular group (community), the requirement on the existence of a fully trusted group manager may not be practical for some applications. Another possible solution is to use undeniable signature [40], where the signer delegates the verification to a third party, and thus signatures cannot be verified without interacting with the third party. For the same reason, it may not be applicable due to lack of the third party in many application scenarios. The ring signature [41] suffers from the same disadvantage as group signature.

As we observe, in our proposed system, the content publisher of content m attaches a signature σ to the content and further caches it in the FCNs for content subscribers to retrieve. We have to guarantee the following two features: 1) the integrity and authentication of the signature associated with the content m, and 2) the privacy of the signature σ.

We plan to apply part of the non-interactive witness-indistinguishable (NIWI) proof system to design our content verification protocol and ensure the above two features. Based on our proposed system architecture, we mainly focus on the following building blocks: *system initiation*, *certificate issuance*, *commitment and proof generation*, and *authenticity verification*. Some of the design ideas are similar to those in [42], [43].

1) *System Initiation*

a) The CRP randomly selects a generator $g \in G_1$ and sets $u_1 := u_2^r \in G_1^2$ and $v_1 := v_2^s \in G_1^2$, where

$u_2 = (g, g^a) \in G_1^2$ and $v_2 = (g, g^b) \in G_1^2$ and $a, b, r, s \in Z_p$ are randomly chosen.

b) The CRP distributes the common reference string $crs := (p, G_1, G_2, e, g, u_1, u_2, v_1, v_2)$ to a valid user.

2) *Signature Generation*

a) The CRP randomly chooses $x \in Z_p^*$ as the secret parameter, and outputs a verification key $k = g^x$ and passes it to the content publisher.

b) For a content $m \in Z_p$, the content publisher outputs a signature [44] as $\sigma = g^{1/(x+m)}$.

3) *Commitment and Proof Generation*

a) The content publisher randomly selects a set of random numbers $\mathbf{r}$, and σ, k, and crs to generate two commitments $\mathbf{c}$ based on σ and k.

b) Similarly, the content publisher also randomly chooses another set of random numbers $\mathbf{t}$, together with σ, k, $\mathbf{c}$ and crs to generate a set of proofs $\boldsymbol{\pi}$.

4) *Authenticity Verification*

a) Given $\mathbf{c}$ and $\boldsymbol{\pi}$, content subscribers can verify the authenticity of σ without knowing the public information about σ and k. The verification equation is given as $e(\mathbf{c}_1, \mathbf{c}_2) = e(g, g)e(u, \boldsymbol{\pi})$.

b) The verification is successful if the following equation holds: $e(\sigma, k \cdot g^m) = e(g, g)$.

Similar to [43], it can be demonstrated that the above signature can securely be verified without revealing its actual value and the public information of the corresponding content publisher. During the content dissemination process, the content publisher caches $\{E(m), \mathbf{c}, \boldsymbol{\pi}\}$ and another set of NIWI proof to prove the equality of the contents in $E(m)$ and in σ of $\mathbf{c}$ to the FCNs so that any public verifier can verify the authenticity and integrity of the content while preserving the privacy of the content and its signature.

In what follows, we give the preliminary simulation and experimental results for our content verification scheme. We use the Pairing-Based Cryptography (PBC-0.5.12) Library and java-based PBC to implement our simulations and experiments. We take Tate pairing as our basic pairing operation. The elliptic curve we use for our scheme is type A. A curve of such type has the form of $y^2 = x^3 + x$. The order of the curve is around 80 bits, as is F_p, the base field, which is considered as the same security level as 1024-bit RSA. For the simulations, we use a laptop with an Intel processor 2.8GHz and 1GB RAM under the platform Ubuntu 11.10. For the experiments, we use a smart phone Nexus S with a Samsung Exynos 3110 processor. The smart phone has 1GHz ARM Cortex A8 core, and 512MB RAM. The experiments are built on the platform Android 2.3.2. All the timing reported below are averaged over 100 randomized runs. As given in Table I, the computational costs for the computer-based platform are 54.6ms, 126.4ms, and 208.9ms for the commitment generation, proof generation, and correctness verification, respectively. For the smartphone platform, the computational costs are much greater than those on the computer platform. Especially, for the verification on the user side, it takes up to 6.199s for verifying a content from another. We can further reduce the computational cost by preprocessing (pp) some computational intensive parts (i.e., pairing operations), as shown in Table I. Note that here the time overhead is per content rather than per packet. Since the size of contents is usually large, we can observe that the time overhead for our content verification scheme is acceptable.

TABLE I
COMPUTATION COST OF CONTENT VERIFICATION IN DATACLOUD

	Simulation	Experiment	Experiment (pp)
Commitment	54.6ms	1.207s	0.498s
Proof Generation	126.4ms	1.316s	0.571s
Verification	208.9ms	6.199s	1.852s

VII. CONCLUSION

Internet of Things (IoT) has emerged as a new paradigm for future Internet. In this paper, we propose a novel ICN-based Internet architecture to enabling advanced data-centric services over IoT. Considering that users in IoT subscribe data-centric services based on their interests, we group the users with common interests together and form various communities in the network. The users within the same community are well connected via communication structure and the community structure is intelligently maintained so that data items can be efficiently disseminated or retrieved. We also discuss mobility management and in-network caching mechanisms under the new architecture. Finally, to protect the data-centric services, we investigate possible security and privacy mechanisms under this newly proposed framework. Our future research will target at the detailed assessment of the proposed architecture, particularly with respect to specific applications, such as mobile health, smart grid energy management, transportation congestion control, etc., over IoT, such that specific design requirements can be addressed accordingly.

REFERENCES

[1] D. Miorandi, S. Sicari, F. Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, September 2012.

[2] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, 2014.

[3] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of iot: Applications, challenges, and opportunities with china perspective," *IEEE Internet of Things Journal*, 2014.

[4] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, 2014.

[5] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework of creating a smart city through internet of things," *IEEE Internet of Things Journal*, 2014.

[6] S. Amendola, R. Lodato, S. Manzari, C. Occhiuzzi, and G. Marrocco, "Rfid technology for iot-based personal healthcare in smart spaces," *IEEE Internet of Things Journal*, 2014.

[7] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe, "A survey of mobility in information-centric networks: challenges and research directions," in *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*. ACM, 2012, pp. 1–6.

[8] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric networking: seeing the forest for the trees," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. ACM, 2011, p. 1.

[9] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 26–36, 2012.

[10] S. Arianfar, P. Nikander, and J. Ott, "On Content-Centric Router Design and Implications," in *Proc. of ACM ReArch 2010*, Philadelphia, USA, November 2010.

[11] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A Data-Oriented (and Beyond) Network Architecture," in *Proc. of ACM SIGCOMM 2007*, Kyoto, Japan, August 2007.

[12] "FP7 PURSUIT Project," http://www.fp7-pursuit.eu/PursuitWeb/.

[13] R. Li and M. Inoue, "A Comparative Survey on Information-Centric Network," IEICE Tenical Report, 2011.

[14] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. of ACM CoNEXT 2009*, Rome, Italy, December 2009.

[15] "NSF Named Data Networking Project," http://www.named-data.net/.

[16] "Content Centric Networking Project," http://www.ccnx.org/.

[17] "FP7 SAIL Project," http://www.sail-project.eu/.

[18] "FP7 CONVERGENCE Project," http://www.ict-convergence.eu/.

[19] C. Dannewitz, J. Golic, B. Ohlman, and B. Ahlgren, "Secure naming for a network of information," in *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*. IEEE, 2010, pp. 1–6.

[20] B. Ahlgren, M. D'Ambrosio, M. Marchisio, I. Marsh, C. Dannewitz, B. Ohlman, K. Pentikousis, O. Strandberg, R. Rembarz, and V. Vercellone, "Design considerations for a network of information," in *Proceedings of the 2008 ACM CoNEXT Conference*. ACM, 2008, p. 66.

[21] R. Li and H. Asaeda, "A community-oriented route coordination using information centric networking approach," in *IEEE Conference on Local Computer Networks (LCN'13)*. IEEE, 2013.

[22] M. Pan, P. Li, Y. Song, Y. Fang, P. Lin, and S. Glisic, "When spectrum meets clouds: optimal session based spectrum trading under spectrum uncertainty," *IEEE Journal on Selected Areas in Communications*, 2014, accepted.

[23] H. Yue, M. Pan, Y. Fang, and S. Glisic, "Spectrum and energy efficient relay station placement in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 5, pp. 883–893, June 2013.

[24] M. Pan, C. Zhang, P. Li, and Y. Fang, "Spectrum harvesting and sharing in multi-hop CRNs under uncertain spectrum supply," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 2, pp. 369–378, February 2012.

[25] W. Ma and Y. Fang, "A pointer forwarding based local anchoring scheme (pofla) for wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 1135–1146, May 2005.

[26] ——, "Two-level pointer forwarding strategy for location management in PCS networks," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 32–45, January–March 2002.

[27] Y. Fang, "General modeling and performance analysis for location management in wireless mobile networks," *IEEE Transactions on Computers*, vol. 51, no. 10, pp. 1169–1181, October 2002.

[28] ——, "Performance analysis of pointer forwarding scheme for wireless cellular networks," in *Proc. 2002 IEEE Global Commun. Conf. (Globecom)*, vol. 3, Taipei, Taiwan, Nov. 2002, pp. 2498–2502.

[29] R. Jain and Y. B. Lin, "An Auxiliary User Location Strategy Employing Forwarding Pointers to Reduce Network Impacts of PCS," in *Proc. of IEEE ICC 1995*, Seattle, USA, June 1995.

[30] J. S. M. Ho and I. F. Akyildiz, "Local Anchor Scheme for Reducing Signaling Costs in Personal Communications Networks," *IEEE/ACM Transactions on Networking*, vol. 4, no. 5, pp. 709–726, October 1996.

[31] Y. Fang, "Movement-based location management and tradeoff analysis for wireless mobile networks," *IEEE Transactions on Computers*, vol. 52, no. 6, pp. 791–803, June 2003.

[32] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. M. Maggs, K. Ng, V. Sekar, and S. Shenker, "Less pain, most of the gain: Incrementally deployable icn," in *SIGCOMM13, August 12C16, 2013, Hong Kong, China.* ACM, 2013.

[33] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005*. Springer, 2005, pp. 457–473.

[34] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 89–98.

[35] A. Fiat and M. Naor, "Broadcast encryption," in *Advances in Cryptology-CRYPTO'93*. Springer, 1994, pp. 480–491.

[36] R. Canetti and S. Hohenberger, "Chosen-ciphertext secure proxy re-encryption," *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 185–194, 2007.

[37] S. Arianfar, T. Koponen, B. Raghavan, and S. Shenker, "On preserving privacy in content-oriented networks," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*. ACM, 2011, pp. 19–24.

[38] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pp. 321–334, 2007.

[39] D. Chaum and E. Van Heyst, "Group signatures," in *Advances in Cryptology - EUROCRYPT91*. Springer, 1991, pp. 257–265.

[40] S. D. Galbraith and W. Mao, "Invisibility and anonymity of undeniable and confirmer signatures," in *Topics in Cryptology - CT-RSA 2003*. Springer, 2003, pp. 80–97.

[41] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Advances in Cryptology - ASIACRYPT 2001*. Springer, 2001, pp. 552–565.

[42] J. Groth and A. Sahai, "Efficient non-interactive proof systems for bilinear groups," in *Proceedings of the theory and applications of cryptographic techniques 27th annual international conference on Advances in cryptology*, ser. EUROCRYPT'08, 2008, pp. 415–432.

[43] L. Guo, C. Zhang, J. Sun, and Y. Fang, "PAAS: Privacy-preserving attribute-based authentication system for eHealth networks," in *The 32nd IEEE International Conference on Distributed Computing Systems (ICDCS'2012)*, Macau, China, June 18-21 2012.

[44] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Advances in Cryptology - EUROCRYPT 2004*. Springer, 2004, pp. 56–73.