AccConF: An Access Control Framework for Leveraging In-network Cached Data in the ICN-Enabled Wireless Edge

Satyajayant Misra†, Reza Tourani†, Frank Natividad†, Travis Mick†, Nahid Ebrahimi Majd‡ and Hong Huang⋆
† Computer Science Department, New Mexico State University, Las Cruces, New Mexico
Email:{*misra, rtourani, fnativid,tmick*}@cs.nmsu.edu
‡ Computer Science Department, California State University, San Marcos, California
Email:{*nmajd*}@csusm.edu
⋆ Electrical and Computer Engineering Department, New Mexico State University, Las Cruces, New Mexico
Email:{*hhuang*}@nmsu.edu

***Abstract*—The fast-growing Internet traffic is increasingly becoming content-based and driven by mobile users, with users more interested in data rather than its source. This has precipitated the need for an *information-centric* Internet architecture. Research in information-centric networks (ICNs) have resulted in novel architectures, e.g., CCN/NDN, DONA, and PSIRP/PURSUIT; all agree on named data based addressing and pervasive caching as integral design components. With network-wide content caching, enforcement of content access control policies become non-trivial. Each caching node in the network needs to enforce access control policies with the help of the content provider. This becomes inefficient and prone to unbounded latencies especially during provider outages.**

In this paper, we propose an efficient access control framework for ICN, which allows legitimate users to access and use the cached content directly, and does not require verification/authentication by an online provider authentication server or the content serving router. This framework would help reduce the impact of system down-time from server outages and reduce delivery latency by leveraging caching while guaranteeing access only to legitimate users. Experimental/simulation results demonstrate the suitability of this scheme for all users, but particularly for mobile users, especially in terms of the security and latency overheads.

***Keywords:* Information-centric networks, threshold secret sharing, authentication, caching, access control, 5G.**

1. Introduction

The nature of the traffic and the service requirements from the Internet have changed tremendously. As per the Cisco Visual Networking Index Forecast (2019) [1]: high bandwidth video traffic would account for 77% of the Internet traffic by 2019 and mobile wireless devices will account for 77% of the world Internet traffic. This implies that the majority of the traffic on the Internet will be multimedia and emanate from wireless mobile users. This immense user generated traffic along with the amount of current and projected machine-to-machine traffic are driving research in domains such as 5G technologies, aimed at providing higher bandwidth and ultra-low latency for mobile applications [2]. This rapid growth has also been fueled by the use of P2P software (Ares, BitTorrent, etc.), which allows each user on the Internet to become a data server. This phenomenon has led to the Internet users becoming indifferent about the data source (video, music, movies) as long as they are reasonably sure about the content. These are alarming signs—the Internet was not engineered to scale for such trends.

This work is supported in part by the U.S. NSF grants:1345232 and 1248109 and the U.S. DoD/ARO grant: W911NF-07-2-0027.

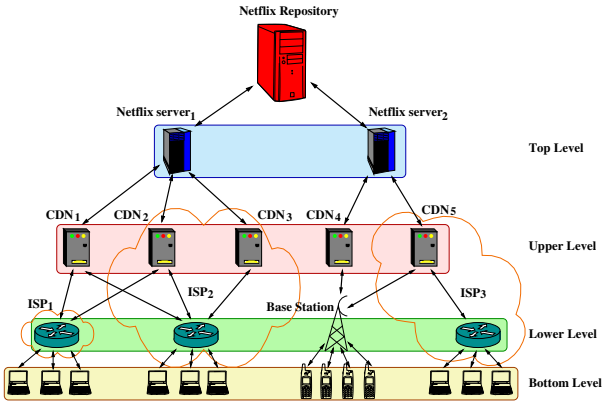


Fig. 1. Multi-level network architecture for Internet-based content distribution.

To address these concerns there has been a strong push to redesign the Internet architecture. This push is aimed at a shift from the *host-centric* Internet to the *information-centric* network [3] where each data item is named and routing is performed using the name. The information-centric networking (ICN) based Internet leverages pervasive in-network data caching and has built-in intelligence to satisfy requests by obtaining the data from network caches or the content provider, and transferring it to the requester(s). Several newly proposed Information-Centric Network (ICN) architectures, such as the CCN/NDN [3], DONA [4], PSIRP [5], PURSUIT [6], and NetInf [7], aim to attain the above objectives. We refer the interested readers to a survey on Information-Centric Networks [8] for more information.

Further, the proposed 5G standard also suggests the use of content caching at the network's wireless edge and device-to-device communication to achieve the low latency requirement of current and future multimedia applications. We believe that the ICN paradigm is a good candidate for the 5G-edge network due to its inherent caching capability. In today's Internet most Content Providers (CPs) use content distribution networks (CDNs) to cache (store) content geographically closer to the users for faster content delivery. As shown in Fig. 1, the Internet hierarchy consists of CPs at the top, followed by the CDNs (e.g. Akamai and Limelight), and then the ISPs (e.g. Comcast, AT&T, and Verizon), culminating in the static/mobile end-users. This architecture places most of the CDN nodes at the edge of ISPs (refer Fig. 1) to reduce the network traffic; yet the ISPs keep deploying more network resources to handle the explosive data growth. The ICN paradigm, with its decoupling of data from the source, will enable in-network caching by the ISPs, reducing their network traffic load and improving scalability and data availability [9]. *But, the important concern is how to ensure that the available cached content are only usable by authentic/legitimate users?*

Let's illustrate this concern using Netflix as the CP and the CCN/NDN Internet architecture [3]. To ensure user authenticity, in the current architecture, a legitimate user's Netflix player authenticates itself to a server hosted on a Cloud service (e.g., Amazon EC2). Once the server authenticates the user, the player/client connects to a CDN node (selected based on network load, proximity, etc.) to access the content. The access control (AC) is enforced by the server and subsequently, streaming happens from the designated CDN node. With ICN, ubiquitous caching would require each node that caches any portion of a content to enforce the AC policies; an impractical exercise. To cope with this problem, the user still has to authenticate herself to Netflix. The decryption key, for the encrypted cached content, is granted to the user upon successful authentication.

However, there is an obvious concern; if the cloud service, Amazon EC2, is down, then the Netflix service is down. The user cannot authenticate herself to use the cached content. This has occurred several times in the past. One may argue that this service-loss can be addressed through better service-level agreements (SLAs) with the cloud provider, but even the best SLA cannot guarantee zero downtime as is evident with many disruptions. Additionally, we note that the ICN paradigm shifts the current tunneling-based security model used by IP networks to a content-based security model, thus securing the content rather than the content-delivery path. Hence, there is a need for an approach that can leverage the contents cached at the routers close to the users, to satisfy requests from legitimate users; otherwise an important facet of ICN cannot be used to improve availability and resilience.

This research is **motivated** by these observations. We address the question: *Can we design an efficient AC framework to utilize the cached content in ICNs that only serves legitimate users/subscribers?* In this paper, we extend our preliminary framework [10] (ACM ICN Workshop, 2013) to answer this question and show that our framework also increases content availability (even when the provider's authentication service is offline) and improves clients' quality of experience.

In a nutshell, our **contributions** include: ***(i)*** Design of ***AccConF*** a *novel* ICN AC framework to guarantee trusted content in nearby caches can be efficiently used by only *legitimate users/subscribers*. AccConF leverages broadcast encryption and specifically targets mobile users that are at the low-end of the devices capability/power spectrum operating at the wireless edge of the network. Our framework also obviates the need for an "always online" authenticator/verifier. ***(ii)*** Discussions on design and implementation issues of AccConF in the popular CCN/NDN architecture. ***(iii)*** Proof that AccConF can handle user revocations limited by a large threshold t and can be augmented to handle more than t revoked users. ***(iv)*** Implementation of AccConF in a CCN/NDN testbed and the ndn-SIM simulator on ns-3 and accompanying analysis validating its usability in mobile devices.

In Section 2, we present the related work. In Section 3, we present the basic definitions and notations, and in Section 4, we present the system model, security assumptions, and the threat model. We present our framework in Section 5, its ICN specific details in Section 6, and discuss its security provisions in Section 7. In Section 8, we present our experimental results and analysis. In Section 9, we present our conclusions.

2. Related Work

In CCN/NDN [3], the user's data interest (request) is either served by an intermediate router that receives the interest and has the data cached or the Content Provider (CP). Data is routed back using information stored in a router's pending interests table and the forwarding information base, and is cached at each forwarding router. In DONA [4], CPs advertise their named content, in form of P:L where P is the hash of their public key and **L** is the content's unique label, to resolution handlers (RHs), which form an inter-domain RH-hierarchy. A user transmits a data request with the help of the RH-hierarchy to a data source, which then transmits the data back along the same path. The data can be cached in the buffer of the involved RHs along the return path. The design paradigms of both PURSUIT and PSIRP [6] involve three separate elements – publishers, subscribers, and the REndezvous NEtwork (RENE) with similar naming scheme as DONA. Rendezvous Points (RPs) in RENE perform rendezvous action between publishers and subscribers and select a path for a publisher/subscriber pair. Network of Information (NetInf) [7] provides a service conceptually similar to the rendezvous service in PSIRP/PURSUIT. *Caching and named data based addressing are integral facets of all these architectures, which are also the only two pre-requisites of our framework.*

AC in the ICN has recently received more attention from the community [11], [12], [13], [14], [15], [16], [17]. In [11], the authors proposed a per-user privacy design in which content chunks are mixed with chunks of cover and the results are published into the network. The user gets the necessary decoding information via a secure back channel from the CP, which requires the CP

to be always online. Fotiou *et al.* proposed an AC enforcement delegation technique [13]. This scheme introduces the Relaying Party (RP) and the Access Control Provider (ACP) entities, which are responsible for storing the content and enforcing the AC policies, respectively. The RP (a caching node) receives the user's request and sends a secret and the corresponding AC policy to both the ACP and the user. The user authenticates himself to the ACP by forwarding the received secret, the policy, and his credentials. The ACP authenticates the user and notifies the RP to transfer data. This technique requires interaction between each router and an ACP, which is not scalable.

Chen [12] proposed a probabilistic encryption-based AC which leverages symmetric/asymmetric cryptographic operations. The proposed mechanism was augmented with a Bloom filter representing the authorized clients' public keys; that is used by the intermediate routers to verify a user before forwarding the encrypted content. In [14], the authors proposed a mechanism built upon name obfuscation and authorized disclosure; the former prevents the unauthorized clients to obtain the content name. The latter requires any entity, with a copy of the content, to perform client authentication and authorization. In this scheme, the content name is encrypted (hashed) to prevent unauthorized access; the content is in plaintext.

Li *et al.* [17] designed a light-weight signature and AC enforcement mechanism that uses per-content tokens. Tokens are generated and assigned by the content provider to the network entities according to their capabilities. Legitimate users authorize themselves to a network router by obtaining the content's private token(s) from the content provider and verifying to the router that they have the token(s). This scheme suffers from the storage overhead of the token (three per content) at the routers and the overhead of token synchronization, which undermine scalability. Also, the mechanism does not scale in the face of user-revocation as it requires complete re-keying at all routers.

Zhang *et al.* [18] proposed a cooperative random interest propagation mechanism to prevent content-client linkability. In this mechanism, routers are clustered into different groups. Upon receiving an interest from a user, the router probabilistically decides to either forward the interest to the upstream router on the path to the provider or propagate it to another router in its/another cluster. This random interest propagation prevents a malicious router from determining whether the interest arrives from a router on the path or from a propagating router.

Attribute-based AC has also been investigated for ICN [15], [16]. In [15] the authors proposed a sketch of the key-policy and the ciphertext-policy based AC. In the key-policy, the content is encrypted with a key that is derived from the content attributes and the access policy is embedded in the decryption key. For the ciphertext policy, the access policy includes the authorized clients' attributes which is used to generate the decryption key. Li *et al.* [16] proposed a ciphertext-policy scheme in which the provider encrypts the content with a symmetric key. It then encrypts the symmetric key with the access policy, which results in the content name. The user first acquires the content name from the name publishing system. Only an authorized user can decrypt the content's name using his attributes to get the symmetric key. The problem with the attribute-based systems is again lack of support for client revocation and computation complexity.

Broadcast encryption (BE) was first proposed by Fiat and Naor [19] to enable a source to send encrypted data to a set of legitimate users in the network who can decrypt the data. The protocol was t-resilient (resilient to collusion of up to t malicious users) with $O = (t^2 \log^2 t \log n)$ message transmission overhead and $O = (t \log t \log n)$ key storage at the user, where t is the revocation threshold and n is the total number of users. In 2001, Naor *et al.* [20] decreased the key storage requirement to $\log n$. Broadcast encryption has found use in the real-world applications. For instance, subset difference based BE is used for AACS, HD DVD, Pay Television, and Blu-ray disc encryption. However, these techniques are not readily usable for secure content delivery on power and computation constrained mobile devices.

In this paper, we extend our preliminary work [10], which uses the public-key based traitor tracing t-resilient algorithm proposed by Tzeng and Tzeng in [21] as a building block to create a secure content delivery framework especially applicable for mobile devices. *Our enhanced framework (AccConF) ensures that mobile devices need less than* 4 *additional seconds at start-up on account of the BE procedures.* This guarantees that user experience is not affected adversely. We also, propose a detailed protocol for handling $|\mathcal{R}| > t$, and address the real-world implementation challenges and present more analysis and experimental results.

Majority of the proposed AC mechanisms in ICN either need an entity (or a network of entities) for client authentication and/or require the intermediate routers to perform client authentication. These assumptions, on one hand, undermine the scalability of the system due to the additional workload of the routers. On the other hand, they undermine the security; in case a router decides to maliciously authenticates an unauthorized client. Different from prior work, in AccConF, there is no entity for AC enforcement; the intermediate routers only forward an extra content (enabling block), which is much smaller than the original content. In the event of client revocation, our framework only incurs a minor updating cost as opposed to the proposed mechanisms, which invariably require system re-keying.

3. Basic Definitions and Notations

From here on, we denote the content provider as CP, the content distribution network as CDN, a CDN node i as CN_i, the Internet Service Provider as ISP. In our framework, the CP and its servers are essentially the same as they perform the same tasks, so we use the terms server and CP interchangeably. A user u_i's public/private key pair for asymmetric encryption/decryption is denoted as $< P_i, Pr_i >$ and the CP's corresponding key pair is denoted as $< P_{\mathcal{S}}, Pr_{\mathcal{S}} >$. Now we define some key concepts used in the paper (please refer to [22] for details). Table I presents the notations used to describe our definitions and framework.

Definition 3.1: [Broadcast Encryption] Broadcast encryption is defined as a mechanism where a CP can securely broadcast content to a set of legitimate users $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$, such that each $u_i \in \mathcal{U}$ can decrypt the content using his private key (or share). [23] □

Definition 3.2: [Shamir's $(t+1, n)$-threshold Secret Sharing Scheme] In this secret sharing, a secret is shared between n users in a way that at least $(t+1) \leq n$ users have to combine their shares to obtain the secret. No combination of users less than $t+1$ ($t+1$ is termed the *threshold*) can decipher the secret. This scheme is implemented with the help of a one-dimensional t-degree polynomial $p_t(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_t x^t$, which can be uniquely determined using any $t+1$ points on the polynomial. A user u_i's share is given by $(x_i, f(x_i))$, where x_i is a point on the X-axis and $f(x_i) = p_t(x_i)$. In Shamir's secret sharing scheme, generally the secret is the term a_0 in the polynomial. □

Definition 3.3: [Lagrangian Interpolation Polynomial] A Lagrange's polynomial of degree n taking on the values $f(x_0), \ldots, f(x_n)$ for the points $x_0, \ldots, x_n$ is given by,

$$L_n(x) = f(x_0)\frac{(x-x_1)(x-x_2)\ldots(x-x_n)}{(x_0-x_1)(x_0-x_2)\ldots(x_0-x_n)} +$$

$$f(x_1)\frac{(x-x_0)(x-x_2)\ldots(x-x_n)}{(x_1-x_0)(x_1-x_2)\ldots(x_1-x_n)} + \ldots +$$

$$f(x_n)\frac{(x-x_0)(x-x_1)\ldots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\ldots(x_n-x_{n-1})}.$$

Note that the secret a_0 in Shamir's secret sharing scheme can be obtained as $a_0 = L_n(0)$. In this paper, we denote the i^{th} fractional term (also called the Lagrangian coefficient) in $L_n(0)$ as, $\lambda_i = \Pi_{0 \leq j (\neq i) \leq n} \frac{x_j}{x_j - x_i}$ resulting in $a_0 = L_n(0) = f(x_0)\lambda_0 + f(x_1)\lambda_1 + \ldots + f(x_n)\lambda_n$. □

With Shamir's secret sharing, when $t+1$ users combine their shares, they can obtain a unique interpolating polynomial using well-known techniques, such as Lagrangian interpolation. The Lagrangian interpolation method uses the Lagrangian Interpolation Polynomial to interpolate $p_t(x)$.

Definition 3.4: [Decisional Diffie-Hellman (DDH) Problem] Let G denote a multiplicative finite cyclic group of order Q (a large prime number), and g be a generator of G, then given two distributions (g^x, g^y, g^{xy}) and (g^x, g^y, g^z), where $x, y, z \in \mathcal{Z}_Q$ ($\mathcal{Z} \setminus Q\mathcal{Z}$), the set of non-negative integers truncated by Q, and are chosen at random, can the two distributions be distinguished? This DDH problem is widely assumed to be intractable [22]. □

Definition 3.5: [Schnorr Group] Given two large primes Q and P, where $P = rQ+1$, $r \in \mathbb{Z}_Q^*$, where $\mathbb{Z}_Q^*$ is the multiplicative group of integers $mod\ Q$, choose $1 < h < P$, such that $h^r \not\equiv 1\ mod\ P$, then $g = h^r$ generates a Schnorr group, which is a subgroup of $\mathbb{Z}_P^*$, the multiplicative group of integers $mod\ P$ of order Q. [24] □

We use the Schnorr group for our framework. In cryptography, such prime-order subgroups are desirable as the modulus is as small as possible relative to Q.

TABLE I
NOTATIONS USED

Notation	Description
P, Q	Big prime numbers such that $P = 2Q+1$
$\mathbb{Z}_Q^*, \mathbb{Z}_P^*$	Multiplicative groups of integers of order Q and P respectively
$\mathbb{G}_Q, \mathbb{G}_P$	Cyclic groups of order Q and P respectively
g	Generator of a sub-group of $\mathbb{G}_P$ of order Q
$ZQrand()$	Random number generator in $\mathbb{Z}_Q^*$
a_0	Constant of $p_t(x)$
t	Degree of polynomial $p_t(x)$ and the revocation threshold
$\mathcal{R}$	Set of revoked users, $\lvert\mathcal{R}\rvert \leq t$
n	Total number of legitimate users
τ	Secret (Symmetric) key for data encryption
$T_i = (x_i, f(x_i))$	Tuple of user u_i
$T_r = (x_r, f(x_r))$	Tuple of revoked user u_r
$\lvert\rvert$	Concatenation operator
$p_t(x)$	One-dimensional t-degree polynomial
$f(x_i)$	Evaluation of coordinate x_i on $p_t(x) \in \mathbb{Z}_Q^*$
E	Server's share (Protocol 1)
E^e	Transformed server's share (Protocol 2)
$\mathcal{S}_C$	Enabling block
γ	Encrypted symmetric key τ
Λ	Set of partial Lagrangian coefficients
Υ	Secret key composed of smaller keys concatenation
λ_k	k^{th} Lagrangian coefficient
$\hat{\lambda}_k$	k^{th} partial Lagrangian coefficient

4. SYSTEM AND THREAT MODELS AND ASSUMPTIONS

In this section, we present the system model, our security assumptions, and possible security threats to our framework.

A. System Model

As the next generation Internet architecture is a notion that is constantly in flux, we model AccConF to be adaptable. The setup is hierarchical as shown in Fig. 1, where the CPs (or their servers) form the top level of the network. The servers may be synchronized to have the same global image of the user base or be distributed, serving non-overlapping user groups, while still having access to the central content repository. The content is pushed onto the CDN nodes–the next (upper) level of the system hierarchy–to transmit the data to the users connected to the ISPs. The next (lower) level consists of ISPs, which cache the packets and forward the data to the users (bottom level).

For illustrating our framework and experimentation, we use the NDN architecture [3], and its code-base CCNx [25]. However, with all ICN architectures sharing the same premise of caching and name based routing our framework will apply to all. In our framework, Q and $P = 2Q+1$ are large prime numbers, n is the number of users in the system, t is the number of users that can be revoked without affecting system performance; and given that all polynomial operations happen in $\mathbb{Z}_Q^*$ (Q is used to define $\mathbb{Z}_Q^*$), n has to satisfy the condition $n \leq Q - t - 1$. We use $r = 2$ for simplicity of exposition. A first-time user registers with the CP to get his credentials and can obtain data from proximal nodes or the CP.

B. Set-up and Security Assumptions

We assume that the content is encrypted by the content provider using a secure symmetric key encryption algorithm, such as

AES [22]. A content or a group of contents (set of movies) may be encrypted using the same secret key–a legitimate user can decrypt the set of contents after successfully extracting the key. Different secret keys can be used by the provider to encrypt different contents or groups of contents; this allows the provider to define diverse AC policies. Our framework's objective is to ensure that the content is encrypted and cannot be used by an entity that is not a legitimate user/client (not even CDN/ISP nodes).

We also assume that a legitimate user's front-end player does not store the symmetric key after decrypting the content, and that a user cannot tamper the player, which performs the decryption. Most content providers (Netflix, DirectTV, Comcast) have a player (set-top box, a standalone or embedded player), which performs the task of decryption of the content and these players are not easily tamperable. Without this assumption, no known encryption scheme can be used for security. We assume that the user does not use VPN tunneling or other location-cloaking mechanisms, such as the Tor network [26], to hide their location. In the rest of the paper, we use the term *user* and *client* to refer interchangeably to the *user's mobile device*.

C. Threat Model

In a set-up for content delivery, data security is of utmost importance. The use of symmetric key infrastructure, public key infrastructure, and our framework guarantees data security. However, there are several other attack scenarios.

(a) For instance, an attacker could flood the network with fake interests (new or replayed interests), thus orchestrating a denial of service (DoS) attack [27].

(b) An adversary can pollute the routers' caches by sending out unpopular requests [28].

(c) Traffic analysis attack can be performed on a specific user to identify his content access pattern [11].

(d) A compromised or colluding user's keying materials can be extracted and used by an adversary, not part of the system, to gain access to the content by impersonation. The extracted keying materials can be used by an adversary to mount a Sybil attack [29].

(e) Also, few revoked users (popularly termed as traitors) can collude to generate a key for a malicious user (pirate), not part of the system, to decode the content.

(f) Additionally, there are standard attacks by an adversary, such as chosen plaintext attack (CPA), chosen ciphertext attack (CCA), and adaptive chosen ciphertext attack (A-CCA) [22], [21].

We refer the interested readers to a comprehensive survey on ICN security [30] for more information on the impact of these threats and proposed countermeasures.

The use of the information-centric paradigm can address some of the threats mentioned above. For instance, the use of the sequence numbers in the interest and data packets, and caching at the edge routers can help neutralize replay attacks. Aggregation of interest packets and controlling interest rates can reduce the impact of DoS attacks. Note that neither the NDN architecture nor our framework require the users to identify themselves to the communicating hosts nor in the interest packets. This ensures identity privacy, unless of course, the routers in the user's neighborhood collude to identify him. Cache pollution attacks have been addressed to a satisfactory level in [28].

After proposing our framework, we will discuss how we can provide security against some of the standard cryptographic attacks, such as CPA, CCA, and A-CCA. We will also propose some potential solutions to improve the security of our framework against Sybil attack and discuss the difficulties of orchestrating collusion attack. However, we note that these proposed solutions cannot thwart extreme Sybil attack scenarios, where our framework can be vulnerable.

5. AccConF: Framework for High Availability and Efficiency in Secure Content Delivery

Now, we present our framework, which helps perform the following for AC in an ICN: (i) Allows ISPs to cache the encrypted content packets at their edge-routers enabling requests for same data from different users with different keys to be served from the cache. (ii) Increases the availability of the content to users by not requiring an initial authentication by an online server. (iii) Ensures that only legitimate users can consume the content, according to the content access policy, and no revoked user can. The protocols in our framework are either implemented at the top (the content servers or CDN nodes) or the bottom levels (end user devices) of the system hierarchy (Fig. 1). There are several BE schemes in the literature [19], [20], [21] and our framework is generic enough to use any BE scheme, which can account for user revocation. However, for ease of illustration in this paper, we use a BE scheme proposed by Tzeng and Tzeng [21], which is a variant of Shamir's secret sharing scheme (Definition 3.2).

In [21], the threshold $t+1$ of Shamir's scheme helps define a revocation threshold of t–the threshold for the number of user revocations permitted without affecting data secrecy. Congruently, we assume n legitimate users in the system and the number of revoked users ($|\mathcal{R}|$) to be at most t ($<< n$); we also propose an enhancement to handle $|\mathcal{R}| > t$. The BE scheme proposed by Tzeng and Tzeng was proved by them to be as hard as the DDH problem [21]. We augment the proposed BE scheme to allow accurate and efficient encryption of the content, and to ensure that contents can only be used by legitimate users, but not by the revoked users.

A. Overview of AccConF

Our framework consists of five steps: The first two steps are performed at the server and are related to encrypting τ, the symmetric key for content encryption. The third and fourth steps are the communication interaction between the client and the server including registration and content retrieval; only the last step is performed at the client. Fig. 2 illustrates the interactions between the provider, routers, and clients and their corresponding protocol execution. In the **first step** (Protocol 1), the server generates a polynomial of degree t, evaluates $n+t$ ($>> t$) number

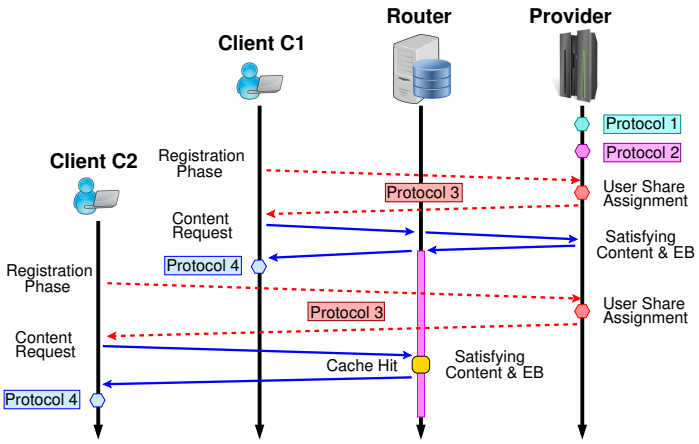


Fig. 2. Communication flow and protocols execution in the network.

of points on it, and keeps t of them as its own shares. The server distributes n of the evaluated points among the n clients, one to each legitimate client. This distribution happens at the time that the client performs registration with the server (Protocol 3).

In the **second step** (Protocol 2), the server generates the enabling block–an essential metadata block, which contains the encrypted τ, and is used by a client in the last step to extract τ. The enabling block is forwarded to the routers similarly as content chunks and forms an integral part of the content. The client needs to register itself with the server in the **third step**. Upon successful registration, in the **fourth step**, the client retrieves the content from the network. In the **fifth step** (Protocol 4), a legitimate client extracts the encrypted τ from the enabling block by using his share.

As depicted in Fig. 2, the provider first generates a polynomial and evaluates the server and clients' shares. Each share is given to a client upon a successful registration; this interaction is shown in Fig. 2 by the dashed red arrows. A router receiving the first request for a content forwards it to the provider for the content and enabling block retrieval as shown by solid blue arrows. Upon receiving the content and enabling block, the router caches and forwards them to the client. The client uses the received enabling block and its share to extract the content decryption key (refer to Fig. 2). Subsequent requests for a cached content retrieve the enabling block and content from the caching router.

B. Basic Protocols

We use a server $\mathcal{S}$ to illustrate the computations at the server(s) or the CP. The server $\mathcal{S}$ generates the polynomial $p_t(x)$ and calculates the tuple $T_i = (x_i, f(x_i))$ for each legitimate user u_i. Where it does not create confusion, in the context of the users, we use share and tuple interchangeably. In what follows, we use index i to represent the users' shares and index j to represent the server shares.

*1) **Polynomial and Shares Generation**:* Protocol 1 presents the procedure for generation of the polynomial $p_t(x)$ of degree t. In Line 1, the server generates the $t+1$ coefficients of $p_t(x)$. It then generates its shares by identifying t random points (Lines 3-5) on $p_t(x)$ and the n clients' shares using n other points (Lines 6-7). The dissemination of the users' share happens through the User Registration Protocol (Protocol 3). The CP encrypts the content using a shared symmetric key $\tau \in \mathbb{Z}_Q^*$. A bigger key (say 128-bit AES key) can also be handled; we will discuss this in Protocol 2.

Protocol 1 Generation of Polynomial/User Shares at the Server

Input: Values of $n < Q$ and t, a prime number Q, $ZQrand()$, and $E = \{\}$.

Output: Generates a polynomial $p_t(x)$ with random coefficients $a_0, \ldots, a_t$ and the tuple T_j for each user u_j.

1: Calculates $a_i = ZQrand()$, $i = 0$ to t.
2: Generates $p_t(x)$ using the a_is.
3: Calculates $x_j = ZQrand()$, $j = 0$ to $t-1$ and $x_j \neq x_k, 0 \leq j,k \leq t-1$. {Ensures x_js are positive, unique, and not reused for clients}
4: Calculates $f(x_j) = p_t(x_j) \in \mathbb{Z}_Q^*$, $j = 0$ to $t-1$.
5: Obtains $E = E \cup (x_j, f(x_j))$, $j = 0$ to $t-1$. {Calculation of each legitimate client's share follows.}
6: Calculates $x_i = ZQrand()$, $i = t$ to $n+t-1$, and $x_i \neq x_k, 0 \leq i,k < n+t-1$.
7: Calculates $f(x_i) = p_t(x_i) \in \mathbb{Z}_Q^*$, $i = t$ to $n+t-1$.
8: Stores values $T_i = (x_i, f(x_i))$. {Tuple of user u_i}

*2) **Generation and Encryption of Enabling Block**:* Protocol 2 deals with the generation of the enabling block, which enables the legitimate user to extract the secret key τ, and is delivered to the user as one of the first content packets. By generating a random number (Line 1), the server obtains the encrypted secret key (γ) using the field generator (g), polynomial constant (a_0), and the secret encryption key (τ) in Line 2. Line 3 shows the transformation of the group generator, g, by an exponentiation operation with the generated random number r. In Line 4, the server calculates Λ (partial Lagrangian coefficients), this precomputed Λ is used at the client for calculating the complete Lagrangian coefficients needed for decryption. As we will show in Section 8 by comparing our framework (Global) with the standard approach in literature (GlobalNP), this partial precomputation step helps reduce the decryption time at the client *tremendously*. Thus, our framework is computation-heavy at the server side, which result in lightweight computations at the clients.

In Line 5, the server calculates the transformed enabling block, obtained by raising g to the power of $rf(x_j)$ $\forall f(x_j) \in E$. In Line 6, the server puts together the enabling block $\mathcal{S}_C$. We will discuss the need for timeout (Line 7) and how to decide a value for TO in the next subsection. The enabling block $\mathcal{S}_C$ is signed by the server (Line 8) to guarantee provenance. A bigger key (say 128-bit key for AES) can be used by splitting the bigger key Υ into smaller sub-keys $\Upsilon = \{\tau_1 || \ldots || \tau_b || \ldots || \tau_m\}$, where each $\tau_b \in \mathbb{Z}_Q^*$ and instead of sending γ, the server can send $\{\gamma_1 = \tau_1 g^{ra_0}, \ldots, \gamma_m = \tau_m g^{ra_0}\}$. The user will combine the split keys

Protocol 2 Generation and Encryption of Enabling Block

Input: Server's share E, $ZQrand()$, $g \in \mathbb{G}_Q$, a_0, data secret key $\tau \in \mathbb{Z}_Q^*$.
Output: Enabling Block $\mathcal{S}_C$
1: Calculates $r = ZQrand()$.
2: Obtains $\gamma = \tau g^{ra_0}$. $\{ra_0 \in \mathbb{Z}_Q^*$ and $\tau g^{ra_0} \in \mathbb{Z}_P^*.\}$
3: Calculates $g^r \in \mathbb{Z}_P^*$.
4: Calculates partial Lagrangian coefficients $\Lambda = \{\hat{\lambda}_k \mid \hat{\lambda}_k = \prod_{0 \le j \ne k < t} \frac{x_j}{x_j - x_k} \in \mathbb{Z}_Q^* \}$.
5: Calculates $E^e = \{\langle x_j, g^{rf(x_j)} \rangle | (x_j, f(x_j)) \in E\}$ for the ts server shares and $rf(x_j) \in \mathbb{Z}_Q^*$ and $g^{rf(x_j)} \in \mathbb{Z}_P^*$.
6: $\mathcal{S}_C = \langle \gamma, g^r, \Lambda, E^e \rangle$
7: Generates a timeout value (TO) for $\mathcal{S}_C$.
8: Sign $\mathcal{S}_C$ using the private key ($Pr_{\mathcal{S}}$) of the server.

to regenerate Υ. This protocol's $\mathcal{O}(t)$ modular exponentiations dominate its running time.

Protocol 3 User u_i's Registration

Input: User's registration credentials.
1: User u_i creates a verifiable profile and successfully enters the system.
2: Server securely transmits the user its public key, $P_{\mathcal{S}}$, its digital certificate, the user's share $(x_i, f(x_i))$, and the expiration time (TO) of the share.

*3) **New User Registration**:* Protocol 3 deals with registration of a new user in our framework. For registration, a user u_i sends a *registration* interest to the CP. The format for the user's name is: $/Netflix/Registration/Unique\_Registration\_ID$. This interest contains u_i's other credentials, encrypted with $P_{\mathcal{S}}$ and signed by Pr_i (u_i's private key). The CP then replies to u_i with a data packet containing u_i's unique valid share encrypted with P_i. The reply is unicast from the CP to u_i and is not cached at intermediate routers. Even if the data is cached by a malicious router communication secrecy cannot be undermined.

*4) **Secret Extraction at the User**:* Protocol 4 presents the procedure used by u_i to extract the secret key (τ) needs to decrypt the content. User u_i verifies the signature of $\mathcal{S}_C$ (Line 1) that he has obtained along with the content. As per Definition 3.3, the k^{th} Lagrangian coefficient λ_k is defined as $\prod_{0 \le j \ne k \le t} \frac{x_j}{x_j - x_k}$, where $0 \le j, k \le t-1$ represent the indices of the server shares and are in $\mathcal{S}_C$; the t^{th} fraction is x_i obtained from user u_i. The server precomputed Λ (part of $\mathcal{S}_C$) is used to obtain complete Lagrangian coefficients, thus reducing the computation time at the client significantly. User u_i simply calculates the last term $(\frac{x_i}{x_i - x_k})$, to obtain the k^{th} Lagrangian coefficient by performing only one multiplication $(\lambda_k = \hat{\lambda}_k \cdot (\frac{x_i}{x_i - x_k}))$, instead of t multiplications (in Line 2). *This precomputation enables the framework's use in mobile devices.*

The following steps obtain the parameters used for the decryption. Line 3 calculates δ_1–the multiplication of the shares $(g^{rf(x_k)}, \forall f(x_k) \in E^e)$ in $\mathcal{S}_C$ raised to their corresponding Lagrangian coefficients (λ_k). The client calculates the Lagrangian coefficient of his share in Line 4 and derives δ_2 through the same procedure as Line 3 (in Line 5). With δ_1, δ_2 and the encrypted symmetric key γ, in Line 6 the client calculates the secret τ. If the secret is Υ (a bigger key), then it can be obtained by a minor extension to Protocol 4: in Line 6, instead of calculating just τ, the user calculates $\{\tau_1, \ldots, \tau_b, \ldots, \tau_m\}$, using the same operations, but using $\{\gamma_1, \ldots, \gamma_b, \ldots, \gamma_m\}$, to recreate $\Upsilon = \{\tau_1 || \ldots || \tau_m\}$. Once the user u_i extracts τ, then she can decrypt the content. This protocol requires $\mathcal{O}(t)$ modular exponentiations–again, the bulk of the running time. We detail the effects of different values of t in the next section.

Protocol 4 Secret Extraction by User u_i

Input: $\mathcal{S}_C$ and T_i, the share of u_i
Output: Secret key τ for data decryption
1: Verifies the signature of $\mathcal{S}_C$ using $P_{\mathcal{S}}$.
2: Calculates Lagrangian coefficient $\lambda_k = \hat{\lambda}_k \cdot (\frac{x_i}{x_i - x_k}) \in \mathbb{Z}_Q^*, \quad (\forall \hat{\lambda}_k \in \Lambda)$
3: Calculates $\delta_1 = \prod_{0 \le k \le t-1} (g^{rf(x_k)})^{\lambda_k}$, where $\delta_1 \in \mathbb{Z}_P^*$ and $g^{rf(x_k)} \in E^e$ contained in $\mathcal{S}_C$.
4: Calculates its Lagrangian coefficient $\lambda_i = \prod_{0 \le j < t} \frac{x_j}{x_j - x_i} \in \mathbb{Z}_Q^*$, where x_i is obtained from T_i.
5: Calculates $\delta_2 = (g^r)^{f(x_i)\lambda_i}$, where $f(x_i)\lambda_i \in Z_Q^*$, $\delta_2 \in Z_P^*$, and $g^r \in \mathcal{S}_C$.
6: Extracts secret key $\tau = \frac{\gamma}{\delta_1 * \delta_2}$.

Theorem 5.1: A legitimate user u_i can use the enabling block $\mathcal{S}_C$ and his own tuple $(x_i, f(x_i))$ and correctly decrypt the secret key τ using Protocol 4. □

Proof: Note that the constant term of the polynomial $p_t(x)$ can be calculated using $L_n(0)$, $a_0 = L_n(0) = f(x_0)\lambda_0 + f(x_1)\lambda_1 + \ldots + f(x_t)\lambda_t$, where the Lagrangian coefficient $\lambda_k = \prod_{0 \le j \ne k \le t} \frac{x_j}{x_j - x_k}$ as per Definition 3.3. Consequently, $ra_0 = r \sum_{k=0}^{t} (f(x_k) \cdot \lambda_k)$, where r is a large random number. By virtue of the fact that g is the generator of the Schnorr group (subgroup of $\mathbb{Z}_P^*$) of order Q, the next few steps follow.

$$\begin{aligned}
\delta_1 * \delta_2 &= \prod_{k=0}^{t-1} (g^{rf(x_k)})^{\lambda_k} \cdot (g^r)^{f(x_i)\lambda_i} \text{(Lines 3\&5 Protocol 4.)} \\
&= g^{r \sum_{k=0}^{t-1} (f(x_k) \cdot \lambda_k)} \cdot g^{rf(x_i)\lambda_i} \\
&= g^{r[(f(x_0)\lambda_0) + (f(x_1)\lambda_1) + \cdots + f(x_{t-1})\lambda_{t-1} + (f(x_i)\lambda_i)]} \\
&= g^{ra_0}
\end{aligned}$$

Hence, $\tau \cdot g^{ra_0} / \{\prod_{k=0}^{t-1} (g^{rf(x_k)})^{\lambda_k} \cdot (g^r)^{f(x_i)\lambda_i}\} = \tau$ and the user can obtain the secret key. ■

6. ICN-Specific Details of AccConF

We now discuss the compatibility of the framework with popular ICN architectures. In publish-subscribe based schemes, such as PURSUIT [6], [5] and NetInf [7], the content's meta-information (number of packets, encoding, etc.) are published

by the CP, whereas, in CCN/NDN or DONA, this information can be elicited by an interest packet sent to the network or the resolution handlers respectively. Our framework requires no extra messages or complexity in the network to leverage data-naming and caching. It can be implemented at every node in the network, including the rendezvous nodes in PURSUIT and the resolution handlers in DONA. Below we discuss the specific design details of our framework from the perspective of the popular CCN/NDN architecture.

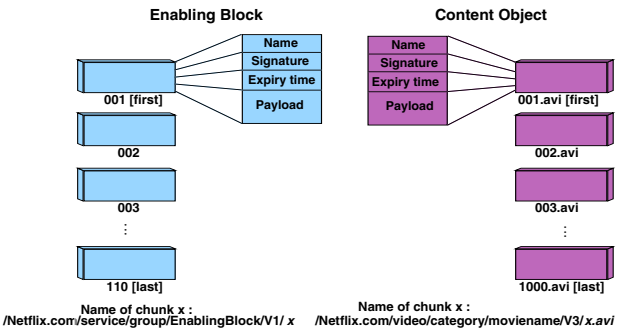


Fig. 3. Our naming scheme for the enabling block and content chunks.

A. Data Chunking and Packet Naming

Large contents are broken down into smaller data packets (chunks); each chunk is named uniquely and requested by its corresponding interest. Fig. 3 illustrates the splitting of the content and the enabling block. Both are split into equal sized chunks and given appropriate names for distinction.

1) Packet Naming: We follow the hierarchical naming convention of CCN/NDN (ref. Fig. 3). A typical content chunk name is */Netflix.com/movie/category/movieName/V3/x.avi*. The first segment is the CP's name, next ("movie") is the content type, followed by the category, e.g, Sci-Fi or Comedy, the fourth is the content name (Star Wars), the fifth is the version (V3), and the last part ($x.avi$) represents the chunk number. Versioning enables coexistence of different content qualities and expiry of content.

The enabling block naming follows the same convention but with the replacement of data type with the service type (premium, standard, plus). In the registration phase, the provider informs the client about the enabling block name according to her service type, which allows the client to request the enabling block corresponding to her service. We note that for better amortized cost a group of content may have one enabling block, so the enabling block name may not include content names. The category segment is replaced by the group with same intention–help group different users under the same service. Two types of numbering scheme can be used: sequential and random.

Sequential Numbering: In this scheme, each content chunk has a sequence number $x \in \{001, 002, \ldots\}$, with increasing value of x. This scheme is easy to implement, but enables cache probing and traffic analysis attack at the router/proxy [31].

Random Numbering: In random numbering, the value of x for the first packet is known to the client; however, each subsequent packet has a random x value. Each chunk carries the sequence number of the next interest to be used. This helps negate traffic analysis attack but, may undermine aggregation of chunks.

B. Protocols to Handle System Dynamics

Our framework has to address several system dynamics. For instance, *(i)* what to do when a registered user discontinues the service and needs to be revoked? *(ii)* What happens when the number of revoked users reaches the threshold t? *(iii)* What happens when a new user arrives and the system is at its capacity? We detail how these events are handled.

1) Revocation of a User u_r: When a user u_r has to be revoked, the server replaces one of its t tuples in $\mathcal{S}_C$ with $T_r = (x_r, f(x_r))$, u_r's tuple. Hence in Line 5 of Protocol 2, one of the $\langle x_j, g^{rf(x_j)} \rangle$ has to be replaced with $\langle x_r, g^{rf(x_r)} \rangle$, thus changing $\mathcal{S}_C$ to $\mathcal{S}'_C$. Several concerns that need to be addressed on this front are: (a) A high rate of revocation would require a new $\mathcal{S}'_C$ to be disseminated in the network with every revocation. Hence, the enabling block should be a small overhead and should be named in a way that allows differentiation between multiple versions in the network. (b) The new $\mathcal{S}'_C$ has to be refreshed everywhere data exists, so that the revoked user cannot access the content.

In Section 8, using implementation results we show that the size of $\mathcal{S}_C$ is much smaller than the content size ($< 1\%$). Also if one key is used to encrypt several contents (e.g., movies), the amortized cost over all related contents can be made negligible. Thus (a) can be addressed. We believe (b) is more difficult to address and attempt some possible solutions.

Lazy Update – Refreshing Enabling Block through Timeout: One way to address (b) is to have a small timeout value (TO) for $\mathcal{S}_C$, which is inversely proportional to the turnover rate ($\zeta(n)$) of users in the system, i.e., $TO \propto 1/\zeta(n)$. The turnover rate is the ratio of the revoked users to all users, per unit of time. This will enforce a small time window in which a revoked user can access the data, after which the routers caching the enabling block will expunge it. Any subsequent request for contents would require a fetch of the latest/updated enabling block.

Proactive Update – Enabling Block refreshed by the CP: Another approach is the CP pushing the enabling block network-wide. Given that the number of users could be as several million spread across the globe, and that the data is cached at several hundred ISPs, this may not be very easy to accomplish. The challenges notwithstanding, such a proactive approach may be feasible with close interactions between the CP, the CDNs, and the ISPs.

Proactive Update – Refreshing Enabling Block through Clustering: An improved approach is to partition the network into independent clusters with number of users $n' < n$, where the clustering is motivated by access policies, geographical distribution, or cluster size. Each cluster C_i has a cluster head (CH), which may be a CDN node or an ISP node, designated by the CP. Each C_i, uses a different polynomial $p^i_{t'}(x)$, and given that

n' could be smaller than n, the threshold t' can also be smaller than t. The enabling block may be generated at the CH or at the CP. In the event of a user revocation, now there is need for only a local update of the local $\mathcal{S}_C^i$ corresponding to the cluster C_i. Updating the routers within the cluster during user revocation becomes much easier. The use of the smaller t' instead of t will also speed-up the user's extraction procedure. A combination of the timeout and the clustering mechanisms may work better than either.

2) Number of Revoked Users Close to or Greater than t: There are two approaches to address this concern:
Proactive Approach: The CP can re-key the whole system with a new polynomial, and treat the already revoked users as non-existent; in essence re-initializing the system. This procedure can be performed when the number of revoked users gets close to t.
Reactive Approach: Let's consider the case where the number of revoked users $|\mathcal{R}| = at + p << n$, where $a > 0$ and $p < t$. Let's assume that the key Υ is 128-bits. As we mentioned in Section 5.B.2, the server splits Υ into m pieces, $\{\tau_1, \ldots, \tau_m\}$. To ensure that revoked users cannot obtain Υ, we can update Protocols 2 and 4 by choosing t revoked users for each $\tau_i \in \Upsilon$ from $at + p$ revoked users, such that each revoked user u_r's share is in the server share for at least one τ_i. Then u_r cannot decrypt one or more τ_is and hence Υ correctly. This can extend the scheme beyond t revoked users.

C. System Reaches User Capacity

The system reaches user capacity when $n + t = \mathbb{Z}_Q^*$. At that point, no new users can be added. All xs are allocated to users, no new unique user share can be created. There is some scope for reuse of the tuples, with the initial revoked users' tuples replaced in the server's share. However, eventually the whole system has to be reinitialized with new prime numbers $Q'(>> Q)$ and $P' = 2Q' + 1$, polynomial, and user tuples and distribution of the new user tuples and enabling blocks. However, we note that this would happen rarely.

7. A Discussion on Security Provisions in AccConF

The security concerns in our framework include Sybil attacks, collusion attacks, and the other well-known attacks, such as CPA, CCA, and A-CCA. We will discuss how the framework can address these concerns. Unfortunately, in an ICN architecture, where routing is based on named data rather than hosts identifiers, there is no way to stop an impersonation attack or a Sybil attack. This is because, if a legitimate user is colluding with an impersonating user (sharing keys, passwords, etc.), then the impersonator has the keying materials of the legitimate node and can decrypt received content. The Sybil attack also follows similar reasoning. As pointed out by Douceur [29], it is difficult to handle such attacks without a central verification entity.

A possible way to identify an impersonating/Sybil node is by the server/CP requiring the user's player to periodically verify its credentials to the server. The verification procedure can involve the CDN node and/or the ISP. Approximate location information obtained during these verifications (from CDN/ISP) can help estimate the user's geographic location (lower the entity in the hierarchy, the finer the localization). A user appearing at multiple locations simultaneously or over a short time span may be part of a Sybil or impersonation attack, and can be revoked. The clustering approach can further limit the impact of the attack. If each cluster uses a different polynomial, then a Sybil attacker using credentials of a user in a different cluster cannot decrypt the data.

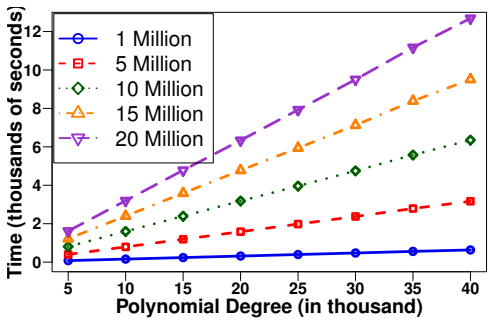
A set of colluding nodes can create a new share for a new malicious (illegitimate) node, however this requires at least $t + 1$ malicious/revoked nodes to collude, armed with the knowledge of Q or P, so that they can re-generate the polynomial using their shares. With $t + 1$ being of the order of thousands (or millions), this is unlikely. Note that the enabling block sent by the server cannot be used to obtain the legitimate shares as it is as hard as the DDH problem. For addressing the other attacks, such as CPA, CCA, and A-CCA, we refer the readers to [21] – the proofs are similar and we omit them here for brevity.

Privacy of the users in an ICN is an important issue, with several privacy threats identified in the literature [32], [31]. The most likely privacy threat is that of cache access monitoring, where an attacker connected to the same router as u_i monitors the cache accesses of u_i to track his behavior. Even though in ICN, especially NDN, user's identity is not present in the packet, an attacker can leverage partial knowledge about the user (e.g., u_i is interested in Sci-Fi movies) and the interest name to conjecture u_i's identity. This problem is being studied by researchers [33] and is not in scope for this paper. We note that if the secret key (τ) is compromised, by means of some attack (a probable event in any secure system), it would require the content to be encrypted with a new key (τ'). Then the enabling block will also need to be updated according to the new key.

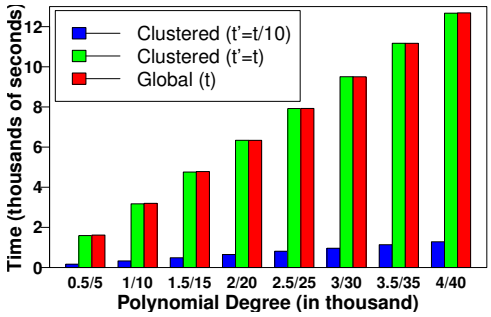
8. Implementation Results and Analyses

Our implementation results are categorized into three segments: *(a)* experiments for performance analysis of our protocols with different settings; *(b)* experiments to assess the cost incurred for providing security (in terms of time) using AccConF over NDN; and *(c)* results from simulation using ndn-SIM on ns-3. Our experiments were implemented on servers, laptops, and Nexus 5 smartphones. All these nodes were a part of a CCNx-0.7 [25] testbed. For the first segment, on the laptop, we implemented our protocols in C (gcc version 4.5.2) and used the GNU Multi-Precision Arithmetic (GMP) library [34] for cryptographic operations. On the smartphones we used Android OS version 5.0.0 (Lollipop) and implemented the application using the Java based Android SDK API-19 (rev. 22.3) Kit Kat and NDK (rev. 9c). Our mobile version was multithreaded and it decrypted the downloaded secret key τ concurrently while receiving content-chunks.

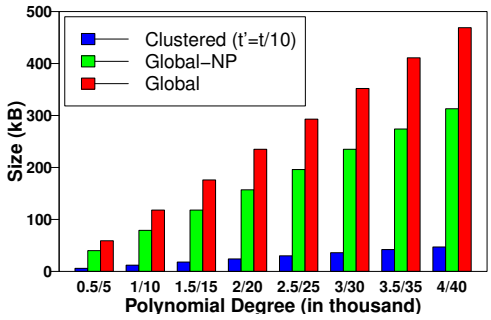
We implemented the Polynomial Generation protocol (Protocol 1), the Enabling Block Generation and Encryption proto-



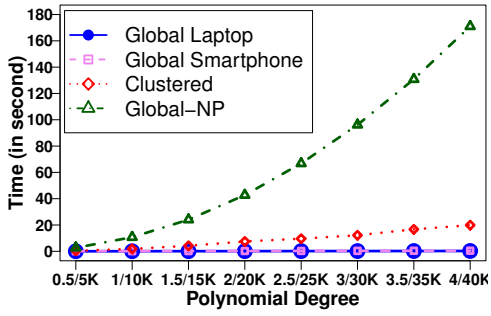
(a) Polynomial and Users Share Generation time (global)



(b) Polynomial and Users Share Generation time



(c) Enabling Block Size



(d) Symmetric Key Extraction

Fig. 4. Results from Protocols Implementation: (a) Time taken to generate $p_t(x)$, $5K \leq t \leq 40K$ (global) and the user shares; (b) Comparison of time taken to generate $p_t(x)$, $5K \leq t \leq 40K$ and user shares in the global and two clustered scenarios ($5K \leq t'(= t) \leq 40K$ and $0.5K \leq t'(= t/10) \leq 4.0K$ respectively); (c) Size of the enabling block for the global and the clustered scenarios ($t' = t; t' = t/10$); (d) Time required for secret key extraction in the global and the clustered scenarios ($t' = t; t' = t/10$).

col (Protocol 2), and the Extraction protocol (Protocol 4). The straightforward user registration protocol was not implemented. In our implementation, for the global scenario, the total number of users ranged from 1M to 20M in increments of 5M, and the value of t ranged from 5K to 40K in increments of 5K, where M and K stand for million and thousand respectively. We chose $n \leq 20$M to represent the dynamic user base of a CP such as Netflix (by current estimates Netflix has $\approx$ 47M US-based and $\approx$ 86M worldwide users) [35]. For the clustered scenario, there were 10 clusters, each having 2M users; we assumed two sets of revocation thresholds (t'): $t' = t$ (as in global) and $t' = t/10$, which ranged from 0.5K to 4K in increments of 0.5K. Protocols 1 and 2, were run on a server class machine with 24 Intel Xeon 2.40 GHz processors and 50 GB RAM. Only one processor was used in our experimental result. Results were averaged over 100 runs.

Fig. 4 displays the results for polynomial and enabling block generations and key extraction. Fig. 4(a) shows the time taken to generate polynomials of different degrees, consisting of generating random coefficients ($\{a_0, \ldots, a_t\}$) for the polynomial ($p_t(x)$), and then evaluating $p_t(x)$ at $n + t$ points. The X-axis represents different polynomial degrees (equivalent to t) and the Y-axis represents the time in thousands of seconds. *The polynomial generation procedure is the most time consuming component of our framework, however, it is executed by the server only and can be performed offline and in parallel by several processors.* We note that the increase in running time with increasing t (for different values of n) is attributable not only to the polynomial degree but also the number of users. The running time scales linearly—the generation time for 20M users is 20 times more than that for 1M users (t being the same).

Fig. 4(b) shows a comparison between the two clustered scenarios ($t' = t$ and $t' = t/10$) and the global scenario on the basis of the polynomial generation time. When $t' = t$, as expected, the time taken is the same. For the $t' = t/10$ case, running time for one polynomial generation is obviously going to be small. In our simulation, generating the ten polynomials of degree t' in parallel took approximately one-tenth of the time needed for generating a polynomial of degree t.

Fig. 4(c) shows the size of the enabling block $\mathcal{S}_C$ in the clustered scenario $t' = t/10$ and two different global scenarios: one in which AccConF is used, thus the partial Lagrangian coefficients are precomputed at the server (Protocol 2), denoted as *Global*, and the other in which no precomputation is performed at the server, denoted as *GlobalNP*. The X-axis represents the polynomial degree (t) and the Y-axis represents size in KiloBytes. The size of $\mathcal{S}_C$ is independent of the number of members; it increases proportionally with t. The worst case size is in Global $\approx$ 470kB, for $t = 40$K. Even then, given that a standard two-hour Netflix movie on a mobile device has a size of around 300 MB [36], the enabling block makes-up less than 0.16% of the movie! For the reactive approach, to handle the case where the number of revoked user $|\mathcal{R}| > t$, the size of the enabling block will increase. Even if we create a share for every byte of a 128-bit secret key (allows $16 \cdot t$ revocations), the enabling block size is 7.52MB–only a 2.5% overhead.

The extra precomputed information at the server results in the

enabling block in Global to be significantly more than that of GlobalNP (around 50% for $t = 40$K). However, as shown in Fig. 4(d), the *corresponding reduction in extraction time at the client due to the precomputation (Global) is significant*–less than 1 second for both the laptop and smartphone versions. The key extraction time in GlobalNP grows super-linearly with increasing t. The laptop client was running on an Apple Macbook Pro running VMware, allocated 1 GB RAM and one 2.5 GHz, Intel Core i5 processor. The smartphone was a Nexus 5, 2.3 GHz quad core, 2 GB smartphone.

As a demonstration of the framework's scalability, especially from the perspective of the expensive extraction protocol, we obtained statistics for higher values of t as shown in Table II. Even when t is 1 million, the enabling block is only 12 MB (1.2% of a standard Netflix movie [35]) and the corresponding extraction of τ takes 1.34 seconds on the laptop and 10.65 seconds on the smartphone. The difference between the laptop and smartphone results are due to the difference in their processors (smartphone's low-power processors are slower). Also, in the laptop the algorithms are implemented in C, while on the smartphones they run on the Java based Android SDK.

Revocation threshold of 1 million is large, as can be seen from recent Netflix statistics [35], [37], reached on an average in three months for Netflix. This makes system re-initialization events rare and scalable. Eventually, a successful/scalable implementation should combine our clustering approach and smaller values of t (say $100,000$), which will allow a smartphone to extract τ in close to 1 second. Such implementations can handle values of n close to 1 million in a cluster. Also, our approach has a modest memory footprint, the high RAM usage numbers (e.g., 143 MB for $t = 1$ million) are only during the extraction process.

For the second results segment, we implemented one client on the Macbook, the smartphone version on the Nexus 5 and the CP (a server with 2.5 GHz Intel Core 2 Quad, 3.8 GB) was five hops away from the clients over a four-tiered network (created using switches and IPv4 routers). We compared the baseline NDN's and AccConF's performance in content retrieval. Our framework took almost the same time for content download, the additional delay being in downloading the enabling block and extracting the secret key—an overhead to enforce the AC. Hence, we define the security cost as the total extra time that it takes for a client to download the enabling block from a nearby cache and extract the secret. Fig. 5 illustrates AccConF's security cost, for different polynomial degrees, for the laptop and the smartphone clients. The cost for the smartphone application increases faster than the laptop's; this can be attributed to the better resources at the laptop's disposal. The biggest cost for the laptop is downloading the enabling block, whereas in the smartphone the costs of communication and extraction are almost comparable. It is interesting that the download time for the smartphone is higher than the laptop despite both connecting to the same access point using IEEE 802.11n. This difference is attributable to the laptop antenna being more powerful than the smartphone antenna.

TABLE II
STATISTICS FOR LARGE VALUES OF t RELATED TO EXTRACTION OF τ

t (in million)	0.1	0.3	0.5	0.7	1
Laptop Extraction Time (secs)	0.14	0.46	0.71	1.03	1.34
Smartphone Ext. Time (secs)	1.16	3.68	5.92	7.44	10.65
Enabling Block Size (MB)	1.2	3.6	6	8.4	12
Smartphone RAM Usage (MB)	12	40	70	96	143

Our last segment details our simulation results using ndnSIM on ns-3. We simulated the AccConF, NDN, and the UDP clients on ten different network topologies; we illustrate the results of four representative ones. The four representative network topologies were: {3755 nodes, 7449 edges}; {3709 nodes, 7193 edges}; {3707 nodes, 7353 edges}; and {3696 nodes, 7331 edges}. The topologies were created using the two-layer Top-Down hierarchical model in BRITE [38]. The autonomous systems (AS) layer was created using the Waxman model and the router layer for each AS was created using the Barabàsi-Albert model. Each topology had two edge routers, each serving five clients through 20Mbps links. One content provider was placed across the network, 6 to 8 hops from the two edge routers. Links in the network core had bandwidth selected randomly between 1 to 4 Gbps.

The server contained 100 content objects. Each object was 300 MB for NDN and UDP and 312 MB for AccConF (12 MB for the enabling block) respectively. The content popularity followed a Zipf-Mandelbrot distribution with $q = 1$ and $s = 2$, which is reflected in the requests made by the clients. The clients constantly requested content–if one content request was satisfied they requested another. For fair comparison, the chunk size was 1436 bytes in NDN and AccConF, comparable to a standard Ethernet frame size. In NDN and AccConF, the routers were equipped with 1.5 GB cache (i.e., 5% of the entire content) and used the LRU cache-eviction algorithm. *We ran the simulation for 30000 seconds.* The simulations were run on a server-class machine having 2 AMD Opteron G276 processors, each core clocking 2.3 GHz, with 128 GB RAM.

Fig. 6 shows the average number of contents downloaded by each client. In NDN and AccConF, the clients' requests are satisfied faster by virtue of nearby caches, hence the clients request more contents. NDN performs a little better than Acc-

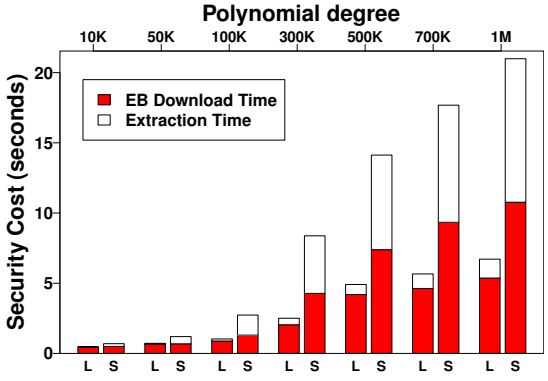
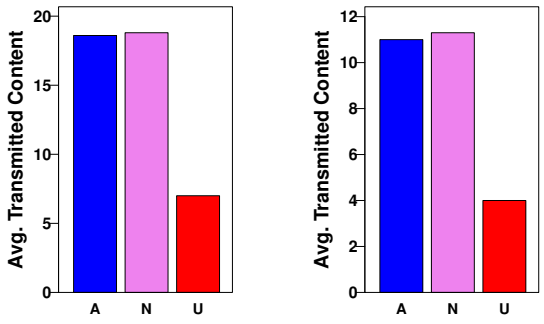


Fig. 5. Security Cost for the Laptop (L) User and the Smartphone (S) User.



(a) Topology 1 (b) Topology 2

(c) Topology 3 (d) Topology 4

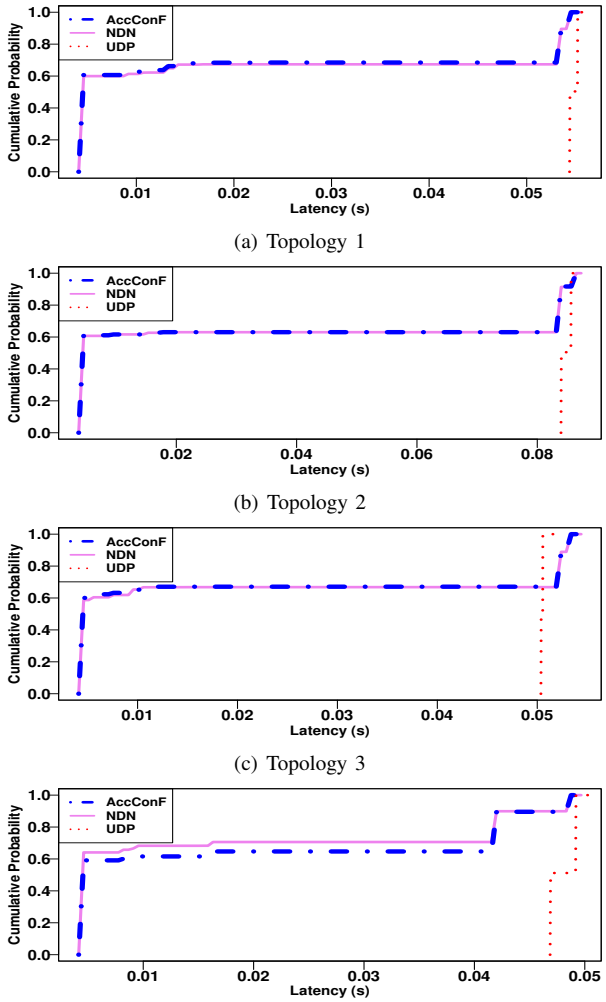
Fig. 6. Average Number of Content Downloaded per client in AccConF (A), NDN (N), and UDP (U).

ConF because it does not have the enabling block. In the last topology, the margin is relatively wider. From our analysis, we identified that this topology's structure is such that more requests are completed, which leads to more cache-evictions and hence more requests being served from farther caches. Consequently AccConF is punished more on account of its enabling block overhead.

Fig. 7 presents the empirical cumulative distribution function (eCDF) for per-interest latencies in the three approaches. NDN and AccConF have a significant number of interests that are served in less than 0.01 seconds, which markedly improves the number of contents downloaded. Fig. 7(d) further illustrates why AccConF has lesser per-client downloads. Whereas in the first three topologies the cumulative probabilities of AccConF and NDN track closely, here AccConF is served by a farther cache (reflected in the eCDF increasing after latency value of 0.042s).

9. Conclusions and Future Work

In this paper, we present a novel access control framework (AccConF) for secure content delivery to legitimate users in ICNs. Leveraging broadcast encryption, AccConF targets the users with power-constrained devices to enable efficient content access without involving an online authenticator. We detailed the protocols and the design decisions for the framework in the CCN/NDN architecture and demonstrated it's feasibility and scalability with practical experiments. Our experimental/simulation results demonstrate that AccConF is practical and deployable with minimal network changes. It can be used by content providers to reduce latency and guarantee high availability of content.



(a) Topology 1

(b) Topology 2

(c) Topology 3

(d) Topology 4

Fig. 7. eCDF for the Latency in AccConF (A), NDN (N), and UDP (U).

In future, we will optimize our smartphone application and the protocols, testing them in a large network. We will investigate more efficient system re-initialization when the system reaches its capacity.

References

[1] Cisco. Cisco visual networking index forecast (2019), 2016. http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/vni-forecast.html.

[2] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung. Cache in the air: exploiting content caching and delivery techniques for 5g systems. *IEEE Communications Magazine*, 52(2):131–139, 2014.

[3] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L. Braynard. Networking named content. In *Intl. conference on Emerging networking experiments and technologies*, pages 1–12. ACM, 2009.

[4] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. *ACM SIGCOMM Computer Communication Review*, 37(4):181–192, 2007.

[5] S. Tarkoma, M. Ain, and K. Visala. The publish/subscribe internet routing paradigm (psirp): Designing the future internet architecture. *Towards the Future Internet*, page 102, 2009.

[6] N. Fotiou, P. Nikander, D. Trossen, and G.C. Polyzos. Developing information networking further: From PSIRP to PURSUIT. In *ICST Conference on Broadband Communications, Networks, and Systems*, pages 1–13, 2010.

[7] C. Dannewitz. NetInf: An information-centric design for the future Internet. In *3rd GI/ITG KuVS Workshop on The Future Internet*, 2009.

[8] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A survey of information-centric networking. *IEEE Communications Magazine*, 50(7):26–36, 2012.

[9] S. Wang, J. Bi, J. Wu, Z. Li, W. Zhang, and X. Yang. Could in-network caching benefit information-centric networking? In *7th Asian Internet Engineering Conference*, pages 112–115, 2011.

[10] S. Misra, R. Tourani, and N. Majd. Secure content delivery in information-centric networks: design, implementation, and analyses. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pages 73–78. ACM, 2013.

[11] S. Arianfar, T. Koponen, B. Raghavan, and S. Shenker. On preserving privacy in content-oriented networks. In *ACM SIGCOMM Information-centric networking (ICN) workshop*, pages 19–24. ACM, 2011.

[12] T. Chen, K. Lei, and K. Xu. An encryption and probability based access control model for named data networking. In *IEEE IPCCC*, pages 1–8. IEEE, 2014.

[13] N. Fotiou, G.F. Marias, and G.C. Polyzos. Access control enforcement delegation for information-centric networking architectures. In *ACM Information-centric Networking Workshop*, pages 85–90, 2012.

[14] C. Ghali, M. Schlosberg, G. Tsudik, and C. Wood. Interest-based access control for content centric networks (extended version). *arXiv preprint arXiv:1505.06258*, 2015.

[15] M. Ion, J. Zhang, and E. M. Schooler. Toward content-centric privacy in icn: attribute-based encryption and routing. *ACM SIGCOMM Computer Comm. Review*, 43(4):513–514, 2013.

[16] B. Li, A.P. Verleker, D. Huang, Z. Wang, and Y. Zhu. Attribute-based access control for icn naming scheme. In *IEE Conference on Communications and Network Security*. IEEE, 2014.

[17] Q. Li, X. Zhang, Q. Zheng, R. Sandhu, and X. Fu. Live: Lightweight integrity verification and content access control for named data networking. *IEEE Transactions on Information Forensics and Security*, 10(2):308–320, 2015.

[18] P. Zhang, Q. Li, and P. P. C. Lee. Achieving content-oriented anonymity with crisp. *IEEE Transactions on Dependable and Secure Computing*, PP(99):1–1, 2015.

[19] A. Fiat and M. Naor. Broadcast encryption. In *CRYPTO*, pages 480–491, 1993.

[20] D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In *CRYPTO*, pages 41–62, 2001.

[21] W. Tzeng and Z. Tzeng. A public-key traitor tracing scheme with revocation using dynamic shares. In *Public Key Cryptography*, pages 207–224, 2001.

[22] A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone. *Handbook of applied cryptography*. CRC, 1997.

[23] M. Naor and B. Pinkas. Efficient trace and revoke schemes. In *Financial cryptography*, pages 1–20, 2001.

[24] C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[25] Palo Alto Research Lab. Ccnx. http://www.ccnx.org/.

[26] Tor Project: Anonymity Online. http://www.torproject.org/.

[27] A. Afanasyev, E. Uzun P. Mahadevan, I. Moiseenko, and L. Zhang. Interest flooding attack and countermeasures in named data networking. In *IFIP Networking Conference*, pages 1–9. IEEE, 2013.

[28] M. Xie, I. Widjaja, and H. Wang. Enhancing cache robustness for content-centric networking. In *IEEE INFOCOM*, pages 2426–2434, 2012.

[29] J. Douceur. The sybil attack. *Peer-to-peer Systems*, pages 251–260, 2002.

[30] R. Tourani, T. Mick, S. Misra, and G. Panwar. Security, privacy, and access control in information-centric networking: A survey. *arXiv preprint arXiv:1603.03409*, 2016.

[31] T. Lauinger, N. Laoutaris, P. Rodriguez, and *et al.* Privacy implications of ubiquitous caching in named data networking architectures. Technical report, Technical Report TR-iSecLab-0812-001, iSecLab, 2012.

[32] A Chaabane, E De Cristofaro, M. Kaafar, and E. Uzun. Privacy in content-oriented networking: Threats and countermeasures. *arXiv preprint arXiv:1211.5183*, 2012.

[33] A. Mohaisen, X. Zhang, M. Schuchard, H. Xie, and Y. Kim. Protecting access privacy of cached contents in information centric networks. In *ACM SIGSAC Symposium*, pages 173–178. ACM, 2013.

[34] The GNU Multiple Precision Arithmetic Library, 2012. http://www.gmplib.org.

[35] Netflix has over 69 million members in over 60 countries., October 25, 2011. http://ir.netflix.com/.

[36] App Makers Worry as Data Plans Are Capped, June 6, 2010. http://www.nytimes.com/2010/06/07/technology/07data.html?_r=0.

[37] OTT Subscriber Annual Churn Rates., July 30, 2015. https://www.parksassociates.com/blog/article/pr-july2015-ott-tracker.

[38] Brite: Boston university representative internet topology generator, 2014. http://www.cs.bu.edu/brite.

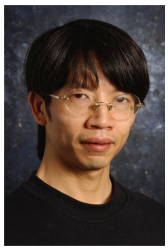
Satyajayant Misra (SM'05, M'09) is an associate professor in computer science at New Mexico State University. He completed his M.Sc. in Physics and Information Systems from BITS, Pilani, India in 2003 and his Ph.D. in Computer Science from Arizona State University, Tempe, AZ, USA, in 2009. His research interests include wireless networks and the Internet, supercomputing, and smart grid architectures and protocols. He has served on several IEEE journal editorial boards and conference executive committees (Communications on Surveys and Tutorials, Wireless Communications Magazine, SECON 2010, INFOCOM 2012). He has authored more than 45 peer-reviewed IEEE/ACM journal articles and conference proceedings. More information can be obtained at www.cs.nmsu.edu/ misra.

Reza Tourani received his B.S. in computer engineering from IAUT, Tehran, Iran, in 2008, and M.S. in computer science from New Mexico State University, Las Cruces, NM, USA, in 2012. From 2013, he started his Ph. D. at New Mexico State University. His research interests include smart grid communication architecture and protocol, wireless protocols design and optimization, future Internet architecture, and privacy and security in wireless networks.

Frank Natividad is currently pursuing his Master degree in the computer science department at the New Mexico State University, Las Cruces, NM, USA. Frank's current interests in research are in power trading agent competitions and machine learning in smart grid.

Travis Mick completed his B.S. at New Mexico State University, Las Cruces, NM, USA in 2014, and is now pursuing an M.S. in computer science at New Mexico State University. His research is in smart grid communication and information-centric networking.

Nahid Ebrahimi Majd received her PhD degree from the department of Computer Science, New Mexico State University, Las Cruces, NM, USA, in 2014. She is currently an assistant professor with the computer science department at the California State University at San Marcos. Her research interest is in energy harvesting wireless ad hoc networks, including relay node placement problem and cooperative caching problem in such networks.



Hong Huang received his B.E. degree from Tsinghua University, Beijing, China, and M.S. and Ph.D. degrees from Georgia Institute of Technology in 2000 and 2002, respectively, all in electrical engineering. He is currently an associate professor with the Klipsch School of Electrical and Computer Engineering at the New Mexico State University. His current research interests include wireless sensor networks, mobile ad hoc networks, network security, and optical networks. He is a member of the IEEE.