LASeR: Lightweight Authentication and Secured Routing for NDN IoT in Smart Cities

Travis Mick, Reza Tourani, and Satyajayant Misra, *Senior Member, IEEE*

***Abstract*—Recent literature suggests that the Internet of Things (IoT) scales much better in an information-centric networking (ICN) model instead of the current host-centric Internet protocol (IP) model. In particular, the named data networking (NDN) project (one of the ICN architecture flavors) offers features exploitable by IoT applications, such as stateful forwarding, in-network caching, and built-in assurance of data provenance. Though NDN-based IoT frameworks have been proposed, none have adequately and holistically addressed concerns related to secure onboarding and routing. Additionally, emerging IoT applications such as smart cities require high scalability and thus pose new challenges to NDN routing. Therefore, in this paper, we propose and evaluate a novel, scalable framework for lightweight authentication and hierarchical routing in the NDN IoT. Our ns-3 based simulation analyses demonstrate that our framework is scalable and efficient. It supports deployment densities as high as 40 000 nodes/km^2 with an average onboarding convergence time of around 250 s and overhead of less than 20 kibibytes per node. This demonstrates its efficacy for emerging large-scale IoT applications such as smart cities.**

***Index Terms*—Information-centric networking (ICN), Internet of Things (IoT), networking, secure onboarding, secure routing, smart cities.**

I. Introduction

THE NEW emerging concept of smart cities applies concepts from the Internet of Things (IoT) to the management of diverse municipal infrastructure and assets [1]. Smart cities will involve large numbers of IoT devices installed in a range of settings from individual homes to critical infrastructure, potentially in a very dense deployment. Considering many of these devices will have limited computational and memory capacities, and will communicate over low-power lossy networks (LLNs), the feasibility of such applications will require advances in efficiency and scalability of IoT networking and communications. Additionally, smart cities will require strong guarantees of security: networked devices will handle large volumes of sensitive information and control valuable assets such as utility infrastructure, thus widening the attack surface for potential compromise. Therefore, strong end-to-end security and privacy mechanisms between smart devices and the cloud are imperative.

Manuscript received May 30, 2017; accepted June 29, 2017. Date of publication July 11, 2017; date of current version April 10, 2018. This work was supported by the NSF under Grant 1719342, Grant 1345232, and Grant 1248109. *(Corresponding author: Satyajayant Misra.)*

The authors are with the Department of Computer Science, New Mexico State University, Las Cruces, NM 88003-8006 USA (e-mail: tmick@cs.nmsu.edu; rtourani@cs.nmsu.edu; misra@cs.nmsu.edu).

Digital Object Identifier 10.1109/JIOT.2017.2725238

Recent literature suggests that information-centric networking (ICN) is a more appropriate approach than Internet protocol (IP) for IoT [2]. Named data networking (NDN) [3], in particular, is a strong architecture for creating scalable and efficient smart city networks, by employing features such as stateful forwarding and in-network caching. In addition, it offers security benefits such as enforced provenance through mandatory network-layer signatures.

Several ICN-based IoT deployments have been announced in the literature, however, no holistic NDN of Things (NDNoT) architecture and protocol suite has yet been proposed. In particular, existing literature tends to neglect concerns related to secure routing and onboarding. Works that do address routing or onboarding do so separately, neglecting the fact that they are closely coupled. As a result, the proposed solutions are limited in scalability, and lack applicability to highly demanding applications such as smart cities. We believe that by exploiting the coupling between routing and onboarding and addressing them *simultaneously*, high degrees of network efficiency and scalability, which are demanded by such applications become achievable.

In addition to introducing a combined approach to routing and onboarding, we employ a hierarchical network structure, a design which has previously been suggested to enable scalability in IoT [4]. Such an architecture allows us to offload much of the burden of routing onto a few less-constrained "anchor" nodes (which may also serve as fog nodes as in [4]), while other devices need only form destination-oriented trees. This approach is similar to that of the IPv6 routing protocol for low-power and lossy networks (RPL) [5], which is currently favored for the IP-based IoT. This is in contrast to previous proposals for the NDNoT, which employed reactive, rather than proactive, routing protocols.

In our framework, secure onboarding is made a prerequisite to routing, in order to help protect the network against routing attacks such as blackholes [6]. Each node in the network is authenticated prior to commencing routing, and in turn a node also authenticates the network it is joining. Since asymmetric cryptography is typically infeasible on IoT devices, we use symmetric cryptography. Our onboarding protocol is based on preshared keys (PSKs) between each node and a designated authentication manager in the infrastructure.

We have combined our approaches to routing and onboarding into a single holistic framework for lightweight authentication and secured routing (LASeR). The combined authentication and onboarding processes are very lightweight, requiring only three round trips and few cryptographic operations.

In summary, the contributions of this paper are as follows.

1) We analyze the current state-of-the-art of routing and authentication in the NDNoT.
2) We propose LASeR, a holistic framework for efficient and secure onboarding and routing in NDN.
3) We demonstrate LASeR's effectiveness and efficiency through analyses conducted in ndnSIM, the NDN module for ns-3.

The remainder of this paper is organized as follows. Section II reviews prior work on NDN and IoT; Section III presents our model for the IoT network and reviews the primitives employed by NDN; Section IV describes the cryptographic materials and operations underlying LASeR's authentication mechanism; Section V presents the protocols employed for onboarding and routing; Section VI offers a simulation-based validation of LASeR's effectiveness; and finally, Section VII concludes this paper and gives an overview of our planned future work.

II. Related Work

Benefits and challenges related to the ICN-based IoT have been previously discussed in [2], [7], and [8], and several architectures have been proposed for both general IoT [2] and specific applications [9]–[11]. However, the majority of these designs focus on service discovery, data delivery, and similar application-centric concerns rather than the initial network bootstrapping or route discovery procedures and their security.

Though most of the aforementioned works do not suggest novel routing protocols for IoT, [2] recognized the routing-related challenges imposed by device constraints in the IoT and proposed a new opportunistic-reactive routing protocol. Under this model, forwarding tables are populated after observing the origins of downstream packets; a flooding-based approach is used as a fallback when no proper route is available. A similar approach was previously outlined in [12].

Other approaches to ad-hoc routing in NDN were reviewed in [13]; the authors identified two broad classes of routing protocols: 1) provider-blind and 2) provider-aware. The provider-blind schemes solely employ controlled flooding to forward requests, while provider-aware schemes add a reactive mechanism like that in the two designs mentioned above.

In addition to the aforementioned NDN-focused routing schemes, some designs have been proposed for other ICN architectures. Among these are cognitive routing frameworks specifically targeting smart city applications, such as [14] and [15], which are unique in that they integrate optimization of quality of information (QoI) into their routing and forwarding procedures.

Bootstrapping and onboarding for the ICN-based IoT have only recently been given serious consideration. Previous architectures such as [11] relied on the asymmetric authentication mechanisms used throughout the NDN stack, however, [16] quantified the time and energy overheads of such schemes on constrained devices and ultimately concluded that their cost is too high. As a result, two designs based on symmetric cryptography were proposed in [17]: a basic implementation of the authenticated key exchange protocol (AKEP2) [18] over ICN, and an improved version which increases its efficiency.

Though [17] efficiently addresses the initial authentication and key-distribution challenges for IoT, it does so without regard to the needs of a routing protocol. As a result, to employ it in conjunction with a separate routing framework would impose additional overhead; the two steps of authentication and routing will occur serially, increasing overhead in the network and overall onboarding latency. In light of this, we propose LASeR, wherein elements of authentication and routing can occur simultaneously to reduce their overall cost.

III. System and Threat Models and Assumptions

In this section, we present the system, network, and threat models and assumptions. For better understanding of our models and assumptions, we start with an overview of NDN.

A. Overview of NDN

The "thin waist" of the NDN stack, as the name implies, is Named Data. In the NDN model, each chunk of data (typically referred to as a *content object*) has a unique *Name*, similar to a uniform resource identifier; the content associated with each Name is typically considered to be immutable. To retrieve a particular content object, a requester sends an *Interest* packet into the network. At a minimum, the Interest contains the Name of the desired content object; it can also contain a signature to verify the requester's identity. The network then retrieves the appropriate content object and delivers it to the requester as a *Data* packet. The Data contains, at a minimum, its Name, the actual content payload, and its publisher's signature. The requester can then verify the signature to ascertain the content object's authenticity. Both Interest and Data signatures typically (but optionally) include a *KeyLocator* field, which contains the Name of the key used for the signature.

Each router in NDN maintains three data structures: 1) a pending interest table (PIT); 2) a forwarding information base (FIB); and 3) a content store (CS). The forwarding procedures for both Interests and Data are based around these tables. Upon receiving an Interest, a router first checks its CS for a match; the CS essentially serves as a cache of Data, indexed by Name. If a match is found in the CS, the Data is served and the request is considered satisfied. If no match is found in the CS, the router then checks its PIT, which indicates whether a previous Interest for the same Name has been forwarded but not yet satisfied.

If a PIT entry exists, the router need not forward the Interest again; instead, it adds the identifier of the incoming interface (*Face*, in the NDN nomenclature) to that PIT entry. If no PIT entry is found, the router consults its FIB (essentially a forwarding table) and employs a configurable *forwarding strategy* to identify the correct Face on which to forward the Interest. The router then adds a new PIT entry indicate that the Interest was forwarded.

Data packets are essentially forwarded following the reverse path as indicated by matching PIT entries. That is, a router receiving a Data checks its PIT to determine the correct Face(s)

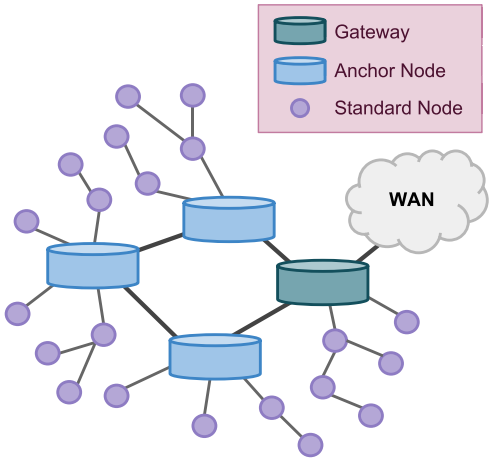

Fig. 1. In our hierarchical island, a gateway connects to the WAN, anchors form an island backbone, and standard nodes form trees rooted at the anchors.

on which to forward the Data. Once the Data is forwarded, the PIT entry is cleared. The Data may then be added to the CS and used to satisfy future requests, depending on the policy employed for cache admission and eviction.

Note that the configurability of the forwarding strategy is an important feature for the application of NDN in IoT. A different strategy can be employed for each Interest depending on its Name prefix, allowing, for example, enhanced quality of service (QoS) depending on the nature of the request. In LASeR, we employ a custom strategy to facilitate our hierarchical network design; more details on this strategy are given in Section V-E3.

B. System Model and Assumptions

We model the NDNoT as consisting of *islands*, which exist at the edge of the greater Internet. The protocols employed within the island need not be influenced by those used in the wide-area network (WAN); therefore, this model is suitable for a local clean-slate deployment of NDN in smart cities prior to wide adoption.

We distinguish between three types of nodes within each island: 1) gateways; 2) anchor nodes (ANs); and 3) standard nodes (SNs). We assume that SNs have small memory, computation, and energy capacities, and employ LLN radios; on the other hand, gateways and ANs are essentially unconstrained. The connections between these entities are visualized in Fig. 1. Gateways serve as edge routers between the island and the WAN, and the ANs are a superset of the gateways and form a backbone or core for the island. Standard nodes wirelessly peer with ANs and use them as sinks to facilitate communication, thus creating trees, or *clusters*, of constituent SNs around each AN.

We assume each SN is assigned a flat identifier (ID), which could either be derived from its media access control (MAC) address or be chosen arbitrarily. For scalability, we will use these IDs to perform routing and forwarding. Nodes can also advertise arbitrary, application-specific Name prefixes; other requesters would then resolve these Names into IDs for the purpose of routing. Namespace creation and management is an NDN- and application-specific decision. This is outside the scope of this paper.

In addition to network entities named above, we assume that there is a service capable of managing the authentication and registration of nodes in the network. We will refer to this entity as the island manager (IM); it may exist either in the cloud, within some particular node, or even as a synchronized database shared between anchors. We assume that the IM and ANs are synchronized to perform secure communication and routing. We do not discuss mechanisms to achieve this, however, it is easy to design and implement. The placement of the IM is an implementation detail which should be made with consideration to the specific needs of a particular deployment. The IM will be responsible for node authentication, and will also serve Name-to-ID resolution requests to support hierarchical routing.

C. Threat Model and Assumptions

We assume that all the devices in the network are capable of performing symmetric key cryptography, such as advanced encryption standard (AES), and message authentication using keyed-hashed functions, such as hashed-MAC. As is standard, we assume that the encryption algorithms and the MAC functions cannot be compromised. In our system, there can be both inside and outside attackers. An outside attacker is not part of the network. It can passively capture data transmissions in the network to perform traffic analysis and also replay captured packets. It can also be an active attacker attempting to masquerade as a legitimate node, and can try to inject false data into the network. An inside attacker is a node that is already on-boarded into the network, it can also inject false data in the network. The false data can include fake route advertisements, enabling sinkhole or blackhole attacks. A compromised or colluding node's keying materials can be extracted and used by an adversary, not part of the system, to impersonate as a legitimate node. This is termed sybil attack; the compromised adversary can operate as a legitimate node in the network. Denial of service and channel jamming can also be threats in our system.

IV. Cryptographic Materials and Primitives

A. Overview

The key hierarchy of LASeR, visualized in Fig. 2, is inspired by that of the PSK extensible authentication protocol (EAP-PSK) [19]. A session between an SN and an IM is identified by the respective IDs of the two parties as well as two nonces (one chosen by each). The SN and IM initially share a PSK, from which two long-lived keys are derived (one for key derivation, one for authentication). With the exchange of nonces and establishment of a session, two additional transient keys are established (one for encryption, one for authentication). These transient keys can be intermittently refreshed simply by exchanging new nonces.

B. Permanent Materials

Each SN is required to store at least two permanent pieces of information: its ID (ID_{SN}) and its PSK (PIN_{SN}), which could

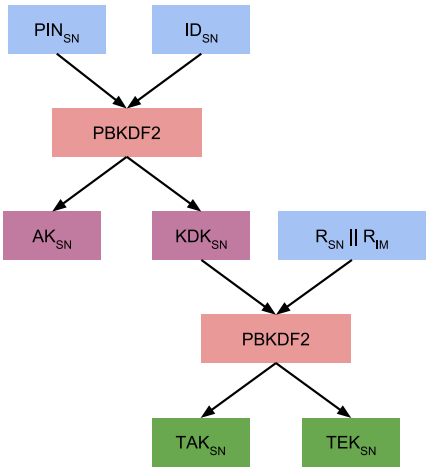

Fig. 2. LASeR's key derivations are based around a PSK and PIN_{SN}. Two permanent keys are shared, and two transient keys are derived per session.

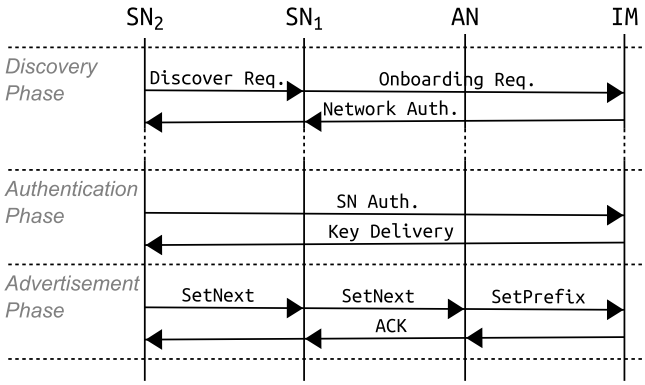

Fig. 3. LASeR consists of three phases, each involving one round-trip to the IM. The discovery phase can be iterated to obtain a desirable path.

be installed at the time of manufacture. The IM is required to permanently store only its own ID (ID_{IM}). The IDs may be arbitrary, and the PSK should be random.

C. Long-Lived Keys

As in EAP-PSK, we use a PSK (in this case, PIN_{SN}) in order to derive two long-lived keys: the authentication key (AK_{SN}) and the key-derivation key (KDK_{SN}). For ease of implementation, we use a password-based key derivation function (PBKDF2) [20], rather than the modified counter mode block cipher used by EAP-PSK, to derive these keys from the PSK. Following the construction in EAP-PSK, we configure PBKDF2 with the following options: PIN_{SN} as the password, ID_{SN} as the salt, and an output length of 256 bits. The first 128 bits of output shall be used as AK_{SN}, and the last 128 bits as KDK_{SN}. We use HMAC-SHA256 as the pseudorandom function behind PBKDF2, due to its wide use and ease of implementation.

The SN may optionally pregenerate and cache both AK_{SN} and KDK_{SN} permanently, though it is not required to. The IM cannot generate these keys until binding time, as it may not have prior knowledge of ID_{SN}. To enable use of NDN's in-stack authentication features, AK_{SN} should be registered on both nodes as `/keys/<ID_SN>/AK`, and KDK_{SN} registered as `/keys/<ID_SN>/KDK`.

D. Transient Keys

To enhance security, the static keys derived directly from the PSK are not used to transmit application data, but only to bootstrap the authentication process. As in EAP-PSK, two nonce-based ephemeral keys will be derived from the key-derivation key. In particular, KDK_{SN} is used to derive two transient keys: a transient authentication key (TAK_{SN}), and a transient encryption key (TEK_{SN}). Again, we use PBKDF2 with HMAC-SHA256 to derive 256 bits of keying material. The key KDK_{SN} is used as the password, and a pair of nonces (R_{SN} and R_{IM}) established during the handshake is used as the salt (details in Section V). These keys are also registered within the local NFD for ease of use: TAK_{SN} is registered as `/keys/<ID_SN>/<R_SN>/<R_IM>/TAK`, and TEK_{SN} as `/keys/<ID_SN>/<R_SN>/<R_IM>/TEK`.

E. Secure Channel

Once TAK_{SN} and TEK_{SN} are derived, a secure channel can be established. Messages are encrypted with AES128-CBC under TEK_{SN}, while TAK_{SN} is used for HMAC-SHA256 signing. These keys are used to deliver an additional symmetric key to used for route advertisement, namely the routing authentication key (RAK).

In the following sections, we use the notation $[\cdots]_{\text{K}}$ to indicate that a message is signed under the key K, and $\{\cdots\}_{\text{K}}$ to indicate that a message is encrypted under K.

V. LASeR Protocol

A. Overview

Onboarding and routing using LASeR occurs in three steps, depicted in Fig. 3: 1) network discovery and authentication; 2) SN authentication and key delivery; and 3) path advertisement. In the first phase, an SN discovers an already-onboarded neighbor, who then asks the IM for the information necessary to authenticate the network to the new SN. In the second phase, the SN authenticates itself to the IM and acquires the keys necessary to advertise a route. The final phase consists solely of the SN advertising its route; the route is then propagated hop-by-hop toward the anchor using SetNext messages. The anchor then notifies the IM of the SN's registration using a SetPrefix message. The full process can be performed in as little as three round trips between a joining SN and the IM.

The resulting routes from SNs to ANs are similar to those which would be obtained by a scheme based on destination-oriented directed acyclic graphs such as RPL [5]. However, each node chooses only one upstream path in LASeR and therefore the result is a forest of trees, each rooted at an AN (equivalent to sinks in RPL nomenclature). Routing between anchors and gateways is assumed to be handled by other means, e.g., a link-state protocol, and is beyond the scope of this paper.

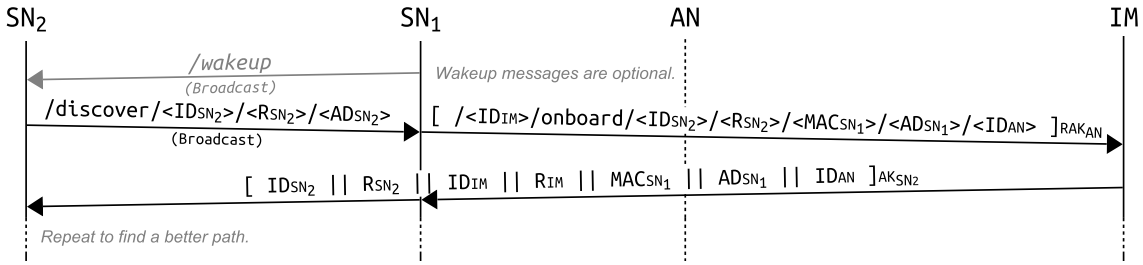

Fig. 4. First phase of LASeR involves SN_2 discovering a network through its neighbor SN_1. The IM authenticates the network to SN_2 using its knowledge of PIN_{SN_2}. The AN (indicated by dotted line) is not involved in the protocol, however, messages between SN_1 and IM will pass through it en route.

B. Network Discovery and Authentication

The first stage of LASeR is network discovery and authentication; that is, an SN discovers a path and verifies the legitimacy of the network it is connecting to. It involves three network entities: 1) the node joining the island (SN_2); 2) its neighbor (SN_1); and 3) the IM. The discovery process may begin either when SN_2 first comes online, or at a later time when it receives a wakeup beacon (a notification from a newly onboarded neighbor that a path to an AN is now available). The complete discovery protocol is presented in Fig. 4.

The first transmission in this phase is a *Discovery Request* sent by SN_2, which constitutes a request to join an island. This transmission is an Interest under the `/discover/` prefix, which is assumed to be broadcast-forwarded in order for SN_2 to identify an immediate neighbor. The Interest should have a relatively long PIT lifetime (likely on the order of minutes), as it may require human input at the IM (to enter PIN_{SN_2}, if it is not preshared) before a Data can be sent in response.

This initial Interest sent by SN_2 contains its ID (ID_{SN_2}), a self-generated nonce (R_{SN_2}), and its current hop-distance from an anchor (AD_{SN_2}, initially ∞); the complete name is `/discover/`$<ID_{SN_2}>/<R_{SN_2}>/<AD_{SN_2}>$. Any neighbor ($SN_1$) which receives this message, is fewer than $AD_{SN_2}-1$ hops from an anchor, and wishes to serve as a relay for SN_2 shall relay it to its AN along with its own MAC (MAC_{SN_1}), its hop-count distance from an anchor (AD_{SN_1}), and the ID of that anchor (ID_{AN}). This message, an *Onboarding Request*, essentially represents SN_1's assent to providing a route toward AN for SN_2. To this end, SN_1 constructs a new Interest for `/`$<ID_{IM}>$`/onboard/`$<ID_{SN_2}>/<R_{SN_2}>/<MAC_{SN_1}>/<AD_{SN_1}>/<ID_{AN}>$ and signs it under RAK_{AN} (which is shared by all successfully onboarded nodes under the AN, as well as by the IM).

Upon receiving this Interest, the IM derives AK_{SN_2} and KDK_{SN_2} according to the procedure outlined in Section IV-C. It generates its nonce R_{IM} and replies with a network authentication message, which is a Data containing ID_{SN_2}, R_{SN_2}, ID_{IM}, R_{IM}, MAC_{SN_1}, AD_{SN_1}, and ID_{AN}. The Data is signed under AK_{SN_2}. This Data authenticates IM to SN_2, informs it of its next-hop neighbor (SN_1), its distance from an anchor (AD_{SN_1} + 1), and its anchor (AN). Because SN_1 changed the Interest name in-flight, it must perform the corresponding reverse mapping in order to deliver the message to SN_2; i.e., the application layer changes the Data's Name from that in the Onboarding Request to that in the original Discovery Request.

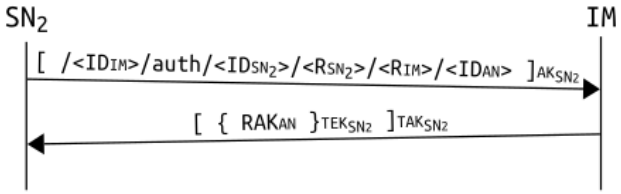

Fig. 5. In the second stage of LASeR, SN_2 authenticates itself to the IM and obtains the RAK corresponding to its anchor.

After obtaining this Data, SN_2 may send a new discover Interest in order to attempt to locate a shorter path to an anchor (in the context of our example, a different node would then take on the role of SN_1). To do so, it sends the same Interest as previously but with a new nonce and an updated AD field. This process may be iterated as many times as desired, or until SN_2 no longer receives a useful response.

When SN_2 is content with its path, it notes its next hop toward the anchor as ID_{SN_1} and its anchor as ID_{AN}, then proceeds to phase two as follows.

C. SN Authentication and Key Delivery

After completing the first phase, SN_2 trusts its island (via its trust for the IM) and is capable of forwarding Interests to any entity within. However, the island does not yet trust SN_2.

In order to establish this trust, SN_2 begins the second phase, which is illustrated in Fig. 5. This phase begins with SN_2 sending its *SN Authentication* (SA), a signed Interest to IM containing the previously exchanged nonces, R_{SN_2} and R_{IM}, as well as ID_{SN_2}, ID_{AN}, and ID_{IM}. This Interest is to be routed using the next-hop information ascertained in the first phase. The IM, upon receiving the Interest, verifies the signature and content and produces a Data packet containing the anchor-specific RAK_{AN} (shared secrets between IM and ANs are always synchronized). The key is encrypted under TEK_{SN_2} and signed under TAK_{SN_2}. At this point, SN_2 is authenticated and can move into the third phase to advertise its path.

D. Path Advertisement

All information necessary for SN_2 to route Interests to other nodes in the island was acquired in the first phase; however, no node is yet able to route Interests to SN_2. In order for

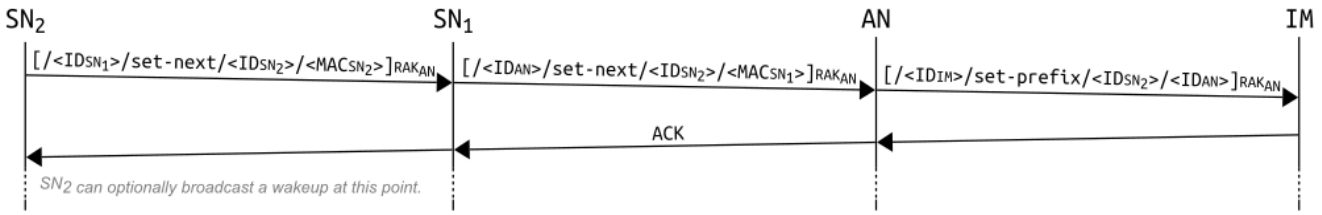

Fig. 6. In the final stage of LASeR, SN_2 notifies its neighbor SN_1 of its commitment to its path. SN_1 then sends a similar notification to the next hop; this repeats until AN learns a route to SN_2. Then, AN informs IM that it is serving as the anchor for SN_2. Note that the IM may be co-located with the AN.

Interests to be delivered to SN_2, each node on the path between SN_2 and AN must know the next hop toward SN_2. To update this routing state, SN_2 sends a notification called a *SetNext* message upstream, signed under RAK_{AN}.

To keep track of downstream nodes, each SN and AN maintains a downstream forwarding base (DFB), which maps a node ID to the next-hop MAC address. The strategy layer of each node uses the DFB and the FIB to make forwarding decisions regarding Interests with destinations in the same AN's cluster.

To inform the next-hop node of its location, SN_2 creates the SetNext Interest with its neighbor's prefix (ID_{SN_1}) and the command `/set-next`, followed by its own ID (ID_{SN_2}) and the MAC address of SN_1's next-hop toward it (in this case, MAC_{SN_2}). This Interest is signed with RAK_{AN}. SN_1 receives this Interest, updates its DFB, then constructs a similar Interest informing the next upstream node that it is the next-hop to reach SN_2. This process, illustrated in Fig. 6, continues until the packet reaches the AN.

When the AN receives this Interest, it updates its DFB and sends a *SetPrefix* notification to the IM to record that it serves as SN_2's anchor. This allows the IM to serve name resolution requests for SN_2. The IM responds with a simple ACK message, which should be forwarded hop-by-hop to satisfy the PIT entries for these Interests, and ultimately notify SN_2 that it has been successfully onboarded.

Upon receiving the ACK, SN_2 may send a wakeup Interest (Name/`wakeup`) to notify nearby nodes that it has been onboarded and can now facilitate their onboarding. This procedure can help expedite the initial onboarding process for an island.

E. Additional Considerations

The above protocols accomplish secure onboarding and routing. In what follows, we will discuss some additional maintenance procedures in LASeR, such as key refresh, prefix resolution, and routing between ANs.

1) Key Refresh: Both the SN's session keys and the AN's RAK may need to be periodically refreshed in order to maintain the security of the island. When the SN wants to change keys, it can either restart from the discovery process, or contact the IM directly to exchange new nonces. In the latter case, the same authentication procedure applies. In order to refresh RAK_{AN}, the IM should generate the new key and send $[\{\texttt{<RAK}_{\texttt{AN}}\texttt{>}\}_{\texttt{TEK}_i}]_{\texttt{TAK}_i}$ to each node i in the AN's cluster, as well as the AN itself.

2) Prefix Resolution: To enable hierarchical forwarding based on ANs, after committing to a path, an SN assumes a new name-prefix rooted under its AN. This prefix is communicated to the IM at the end of the path-advertisement protocol (Section V-D). The IM stores a mapping of IDs to prefixes, and can respond to Interests querying for the prefixes of registered nodes (e.g., respond to an Interest $/\texttt{<ID}_{\texttt{IM}}\texttt{>/get-prefix/<ID}_{\texttt{SN}}\texttt{>}$). Similarly, the SNs and the ANs need to be able to remap Interests onto the appropriate prefixes. Additional arbitrary Name prefixes that a node wishes to serve should also be communicated to the IM. The IM also resolves requests for these arbitrary Names into routable Names, constructed from the corresponding IDs and AN prefixes. The processes for name registration and resolution are quite simple and can be done in various ways. Due to space constraints this discussion is omitted in this paper.

3) Routing Within and Between Clusters: In LASeR, anchors obtain routes to each other out-of-band, for example, using a link-state algorithm such as NLSR [21]. By virtue of hierarchical naming and prefix-based forwarding, anchors need not advertise the constituent nodes in their clusters, only their own prefixes. We assume that the gateway is also a member of this link-state session, and therefore any standard node can reach the gateway just as easily as it can reach an anchor.

In order to reach a node within the same cluster, forwarding nodes use their DFBs. However, each node only has DFB entries for those nodes that rely on it for a path to the anchor. Therefore, in our framework nodes route toward the anchor by default if the DFB lookup fails. As a result, packets between nodes in the same cluster (e.g., machine-to-machine flows) climb the tree toward the AN until the first common ancestor is reached; the packet is then forwarded down the tree to the destination.

To reach a node in another cluster, the IM must first be queried to obtain the node's prefix, as mentioned in Section V-E2. This operation can be made transparent to SNs by offloading it to the ANs. Then, SNs would require no information about prefixes and would incur no penalty of increased transmission burden for prefix lookup. ANs could cache these results to eliminate lookup latency for subsequent requests.

F. Security Analyses

The goal of LASeR is to secure routing in a smart city IoT network. Therefore, we focus on analyzing concerns related

to the authenticity and provenance of obtained routes. We will also remark on privacy concerns where applicable.

The authentication and key exchange procedures in LASeR are effectively equivalent to those of EAP-PSK and AKEP2, which have been proven in literature to be secure. However, the use of a shared RAK for securing routing messages between nodes in a cluster is a potential vulnerability. Though the RAK is always transmitted in an encrypted form, its compromise (through brute force or node takeover) would allow an attacker to publish fake routes on behalf of any node in the compromised cluster.

The result of such an attack would essentially be denial of service of requests destined to that node (assuming the flow's own data is authenticated). This attack can cause blackhole, sinkhole, or wormhole attacks. The LASeR protocol could be augmented to report any changes in the topology to the IM, giving it complete knowledge of the network. Sophisticated algorithms for detecting blackholes, sinkholes, or wormholes can then be employed at the IM. Once detected, the compromised SNs can be revoked and the RAK can be securely refreshed for the remaining legitimate SNs.

In many IoT onboarding protocols, the compromise of an already-trusted node can have major impacts. In LASeR, this would result only in the attacker gaining knowledge of the RAK and the node's own PSK, which it can use to inject fake routes as described above; any application-specific information obtained would not impact routing security. If the node is identified, its PSK can be de-authenticated by the IM and the RAK refreshed as usual. Any unencrypted data (such as IDs) collected by the compromised node do not undermine the security of the network; however, they may be used for traffic analysis and privacy attacks. Ephemeral IDs and pseudonyms can be used to prevent such information leakage.

Some information about the network topology is leaked by LASeR—a passive attacker could determine the layout of the network by observing Onboarding Request packets en route to the IM. Though we have chosen not to prioritize the protection of this information, the Onboarding Request and its reply could easily be encrypted under an additional key derived from the KDK. Similarly, the RAK can be augmented with an additional key to enable the encryption of SetNext and SetPrefix messages. Node anonymity can also be compromised by observing IDs in transmission; however, because we allow IDs to be chosen arbitrarily they can be made ephemeral to thwart related attacks.

Channel jamming and other link-layer or physical-layer denial-of-service attacks are beyond the scope of our framework.

VI. Simulation Evaluation

A. Scenario Configuration

Our initial validation of LASeR was done in ndnSIM [22], an ns-3 extension implementing NDN. The implementation of LASeR involves an application-level controller, a custom forwarding strategy, a modified PIT, and a modified Face which supports ad-hoc forwarding. We have also implemented a hop-by-hop fragmentation and reassembly protocol similar to that in NDNLP [23].

We used the LrWpanNetDevice to model 802.15.4 radios with slotted CSMA/CA, with the *Log Distance Propagation Loss* and *Constant Speed Propagation Delay* models to simulate the radio channel. Nodes were configured with a transmit power of 0 dBm and receive sensitivity of -106.58 dBm, providing an effective range of about 80 m.

Unfortunately, the practicality of executing large-scale wireless scenarios in ns-3 is limited (as interference calculations become prohibitively expensive), so we focus on a cluster of SNs around a single AN in each experiment. The cluster forms the building block of any IoT network, which is essentially composed of several such clusters. Onboarding in a single cluster is representative of that in all clusters, as they can happen in parallel. This is especially true if each AN operates on a separate channel and thus interference between them is minimized. Therefore, we have not modeled the interconnections between anchors, nor a gateway to a WAN, and we assume that each anchor is capable of acting on behalf of the IM. We do not implement the IM's prefix resolution service for this validation study.

In our evaluations, we explore two settings: 1) increasing density of nodes within a fixed area and 2) decreasing density of a fixed number of nodes. Two sets of scenarios were created for these two settings; they will be detailed in their respective sections. For each scenario, we will explore *four statistics*: 1) time for onboarding convergence; 2) the transmission burden of each node; 3) the size of each node's subtree; and 4) the hop-count distances between nodes and their anchors. Convergence times and hop counts serve as indicators of scalability, while transmission burdens and subtree sizes correspond to the energy efficiency of the protocol.

B. Increasing Density

To study the effects of increasing node density, we created scenarios wherein varying numbers of SNs are placed uniformly at random in a 50×50 m^2 (0.0025 km^2), and a single AN is placed at the center. We evaluated scenarios of 40 nodes, 60 nodes, 80 nodes, and 100 nodes; this corresponds to densities ranging from 16 000 to 40 000 nodes/km^2. For each scenario, we averaged results over 20 runs with different pseudo-random number generator seeds; note that the seed affects both node placement and network behavior. To simulate real-world deployments the SNs power-on at random times in the network. The time follows an exponential distribution with $\lambda^{-1} = 120$ s (2 min). An SN attempts to join the network after it is powered-on.

1) Convergence Time: Fig. 7(a) depicts the empirical cumulative distribution functions (eCDFs) of network convergence times; the *x*-axis represents time, while the *y*-axis gives the cumulative proportion of nodes that have been onboarded. As SNs power on randomly, we identified that congestion is not a big challenge to onboarding—new nodes get onboarded rapidly. No clear trend can be seen between the 40, 60, and 80 node scenarios. However, the 100-node scenario clearly converges slower, suggesting that as density increases, radio

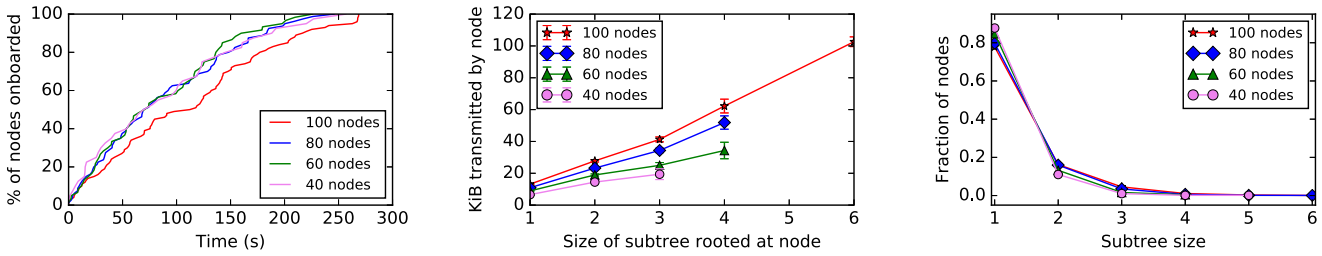

Fig. 7. Simulation results for the four scenarios with increasing deployment density within a 50 × 50 meter area. (a) Empirical CDFs of node convergence time. (b) Transmission burdens by subtree size, with standard-error-of-mean. (c) Probability mass function of subtree sizes.

interference has increasingly adverse impact on onboarding. We believe that with greater densities, onboarding may not converge. The worst-case convergence time across all runs was 314.6 s (5 min, 14.6 s), not much longer than the average case for 100 nodes, 271.0 s (4 min, 31.0 s). We believe this to be an acceptable convergence delay, as the process only occurs once.

2) Transmission Burden: The amount of energy consumed by a node is dominated by wireless transmissions. In LASeR, the transmission burden of an SN grows as it serves increasing number of other SNs as a forwarder in the onboarding process. Therefore, we evaluate the transmission burden observed for SNs of varying subtree sizes. Application-introduced bandwidth is not considered here; only LASeR traffic is measured. Anchor nodes are not included in this analysis, as we assume they are not subject to power constraints.

Fig. 7(b) summarizes our analysis of this transmission burden under increasing subtree size and network node density; subtree sizes are on the *x*-axis, and total kibibytes (KiB) transmitted is on the *y*-axis. A clear linear trend is visible under increasing subtree size; as expected, transmission burden is approximately proportional to the number of nodes being served in the subtree. Additionally, notice that increasing node density results in overall larger transmission burdens, a pattern resulting from interference-related retransmissions.

In our simulations, a single node must transmit an average of 9.89 KiB throughout the onboarding process; while this is a reasonable burden, we notice that this burden is compounded by increasing the number of downstream SNs. For this reason, care must be taken to avoid creating large subtrees in a real IoT deployment.

3) Subtree Size and Distance From Anchor: Fig. 7(c) depicts the empirical probabilities of each observed subtree size; the *x*-axis gives subtree sizes, and the *y*-axis gives the likelihood that a node would host a subtree of that size. On average, 83.6% of nodes served no children, and thus would experience the minimum transmission burden. In accordance with intuition, hop-count distance from an anchor was correlated to subtree size; on average, 79.2% of nodes were only one hop away from the anchor. However, we observed that increasing node density increased the average subtree size and path length, even though the nodes' physical distances from the anchor were unchanged (80 and 100 node cases lead to size six subtrees). This is because increased density increases interference, thus reducing the effective transmission range of nodes and increasing the chance of SNs to use intermediate forwarders to reach the AN. But, we note that the average path-lengths between SN and AN was no more than six hops in any of our scenarios.

C. Increasing Distance

In the previous section, we focused on increasing the number of nodes deployed in a fixed 50 × 50 m^2 area. We now evaluate a second set of scenarios, wherein the node count is fixed at 100 and the deployment area is varied. This increases sparseness, causing creation of longer paths from SNs to the AN and bigger subtrees. We chose areas of 50 × 50 m^2 (0.0025 km^2), 100 × 100 m^2 (0.01 km^2), 200 × 200 m^2 (0.04 km^2), and 400 × 400 m^2 (0.16 km^2). Again, results are averaged over 20 runs. The time at which nodes come online is exponential with $\lambda^{-1} = 120$ s. With a few exceptions, trends are similar to those observed in the fixed-area scenario set.

1) Convergence Time: The convergence times under the 100-node scenarios are visualized as eCDFs in Fig. 8(a). We again see that the densest scenario reaches final convergence slowest, however, it is notable that the sparsest scenario (0.16 km^2) has a slower initial progression. An inflection point is visible between 100–120 s, suggesting that connectivity is poor prior to a sufficient number of nodes (which will serve as intermediate forwarders) being onboarded.

2) Transmission Burden: The transmission burdens for nodes with each observed subtree size are visualized in Fig. 8(b). Again, there is a clear trend of increased burden under higher densities, when considering similarly sized subtrees; this is due to interference-related retransmissions. However, we can see that in the sparsest scenario, some nodes host much larger subtrees and thus carry a greater burden. The available energy at these nodes serves as the bottleneck for communication from the downstream nodes. Thus, care must be taken in node placement: nodes' distances from anchors should be minimized.

3) Subtree Size and Distance From Anchor: The observed probability mass function of subtree sizes is given in Fig. 8(c). Again, a majority of nodes host no children; however, in the sparsest case, a few nodes hosting large subtrees are observed. The emergence of large subtrees can be averted by careful node and anchor placement. The distributions of hop-counts in the three densest scenarios are similar to those observed in the fixed-area scenario set. However, in the sparsest scenario a majority of nodes are two hops from an anchor, rather than

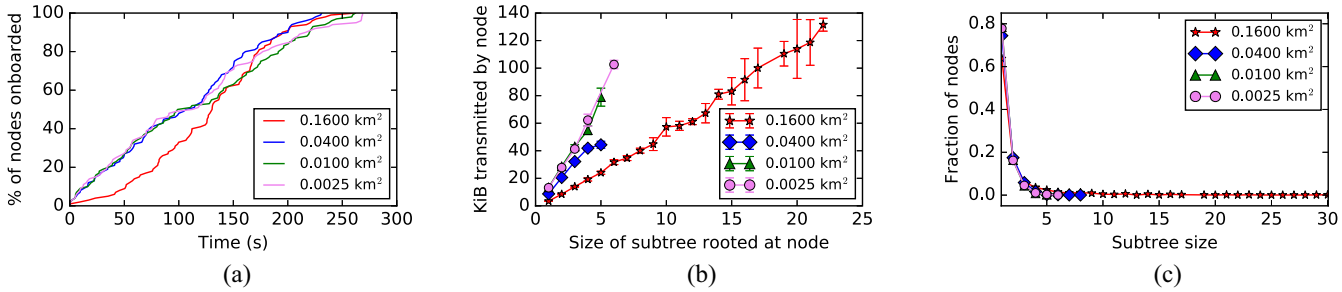

Fig. 8. Simulation results for the four scenarios of 100 nodes with increasing deployment area. (a) Empirical CDFs of node convergence time. (b) Transmission burdens by subtree size, with standard-error-of-mean. (c) Probability mass function of subtree sizes.

one hop. This increases latency; thus, it is best to have short paths from SNs to their corresponding AN.

D. Discussion

The results we have presented demonstrate that LASeR performs well in a wide range of deployment conditions at densities up to 40 000 nodes/km^2. However, as previously mentioned, additional ANs can be deployed on alternate channels. With 802.15.4's 16 nonoverlapping channels, LASeR could scale to well over 500 000 nodes/km^2; therefore, it is a feasible solution for large-scale smart cities.

As explained in Section II, previous approaches to routing in the NDNoT rely on Interest flooding. Thus, each node in the network could potentially be burdened by each Interest broadcast. Though reactive mechanisms help reduce the burden for subsequent Interests to successfully routed prefixes, these protocols are still vulnerable to flooding attacks wherein Interests with unroutable names could be issued to force spurious broadcasts and thus drain energy from SNs.

Furthermore, link-layer ACK (or an emulation thereof in NDNLP) is not possible for broadcast packets, due to the sheer number of replies which would be generated for each broadcast. As a result, these reactive schemes would be forced to rely on application-layer retransmission if a broadcast frame containing an Interest is lost. This would negatively impact QoS, especially in smart-city scenarios where real-time applications are common. We note that this is true for non-ICN based IoT approaches as well.

In contrast to the reactive schemes, LASeR utilizes broadcast only for initial neighbor discovery, and only those nodes on the path to the AN are burdened by forwarding of other types of packets. Therefore, it is not subject to these detriments.

VII. Conclusion

In this paper, we have proposed LASeR, a secure onboarding and routing framework for NDN-based IoT networks. Scalability is achieved through a hierarchical network design, and very little cryptographic or computational burden. Evaluation by simulations confirmed that LASeR requires minimal network overhead and achieves acceptable onboarding convergence times.

The current implementation of LASeR routes based on node IDs, however, an extension is planned to support the advertisement of arbitrary name prefixes. A mechanism to address node mobility with low overhead is also in development. After further validating LASeR in ndnSIM, we intend to implement it on real IoT devices for a live testbed deployment.

References

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[2] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch, "Information centric networking in the IoT: Experiments with NDN in the wild," in *Proc. 1st ACM Conf. Inf. Centric Netw.*, Paris, France, 2014, pp. 77–86.

[3] L. Zhang *et al.*, "Named data networking (NDN) project," UCLA, Los Angeles, CA, USA, Tech. Rep. NDN-0001, 2010.

[4] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the Internet of Things," in *Proc. ACM SIGCOMM Workshop Mobile Cloud Comput.*, Hong Kong, 2013, pp. 15–20.

[5] T. Winter *et al.*, "RPL: IPv6 routing protocol for low-power and lossy networks," Internet Eng. Task Force, Fremont, CA, USA, RFC 6550, 2012.

[6] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, "A survey of routing attacks in mobile ad hoc networks," *IEEE Wireless Commun.*, vol. 14, no. 5, pp. 85–91, Oct. 2007.

[7] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, "Named data networking for IoT: An architectural perspective," in *Proc. Eur. Conf. Netw. Commun.*, Bologna, Italy, 2014, pp. 1–5.

[8] S. K. Datta and C. Bonnet, "Integrating named data networking in Internet of Things architecture," in *Proc. IEEE Int. Conf. Consum. Electron. Taiwan*, 2016, pp. 1–2.

[9] R. Ravindran, T. Biswas, X. Zhang, A. Chakraborti, and G. Wang, "Information-centric networking based homenet," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag.*, Ghent, Belgium, 2013, pp. 1102–1108.

[10] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, "Information centric networking in IoT scenarios: The case of a smart home," in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 648–653.

[11] J. Burke, A. Horn, and A. Marianantoni, "Authenticated lighting control using named data networking," UCLA, Los Angeles, CA, USA, Tech. Rep. NDN-0011, 2012.

[12] M. Meisel, V. Pappas, and L. Zhang, "Ad hoc networking via named data," in *Proc. ACM Int. Workshop Mobility Evol. Internet Archit.*, Chicago, IL, USA, 2010, pp. 3–8.

[13] M. Amadeo, C. Campolo, and A. Molinaro, "Forwarding strategies in named data wireless ad hoc networks: Design and evaluation," *J. Netw. Comput. Appl.*, vol. 50, pp. 148–158, Apr. 2015.

[14] G. T. Singh and F. M. Al-Turjman, "A data delivery framework for cognitive information-centric sensor networks in smart outdoor monitoring," *Comput. Commun.*, vol. 74, pp. 38–51, Jan. 2016.

[15] G. Singh and F. Al-Turjman, "Learning data delivery paths in QoI-aware information-centric sensor networks," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 572–580, Aug. 2016.

[16] M. Enguehard, R. Droms, and D. Rossi, "On the cost of secure association of information centric things," in *Proc. ACM Conf. Inf. Centric Netw.*, Kyoto, Japan, 2016, pp. 207–208.

[17] A. Compagno, M. Conti, and R. Droms, "OnboardICNg: A secure protocol for on-boarding IoT devices in ICN," in *Proc. ACM Conf. Inf. Centric Netw.*, Kyoto, Japan, 2016, pp. 166–175.
[18] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Proc. Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, 1993, pp. 232–249.
[19] F. Bersani and H. Tschofenig, "The EAP-PSK protocol: A pre-shared key extensible authentication protocol (EAP) method," Internet Eng. Task Force, Fremont, CA, USA, RFC 4764, 2007.
[20] B. Kaliski, "PKCS #5: Password-based cryptography specification version 2.0, Internet Eng. Task Force, Fremont, CA, USA, RFC 2898, 2000.
[21] A. K. M. Hoque *et al.*, "NLSR: Named-data link state routing protocol," in *Proc. ACM SIGCOMM Workshop Inf. Centric Netw.*, Hong Kong, 2013, pp. 15–20.
[22] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2.0: A new version of the NDN simulator for NS-3," UCLA, Los Angeles, CA, USA, Tech. Rep. NDN-0028, 2015.
[23] J. Shi and B. Zhang, "NDNLP: A link protocol for NDN," Univ. Arizona, Tucson, AZ, USA, Tech. Rep. NDN-0006, 2012.

Travis Mick received the B.S. degree in computer science from New Mexico State University, Las Cruces, NM, USA, in 2014, and is currently working toward the M.S. degree in computer science at New Mexico State University.

His research is focused on information-centric networking, including security and privacy concerns, caching and forwarding strategies, and applications within the Internet of Things.

Reza Tourani received the B.S. degree in computer engineering from IAUT, Tehran, Iran, in 2008, the M.S. degree in computer science from New Mexico State University, Las Cruces, NM, USA, in 2012, and is currently working toward the Ph.D. degree 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.

Satyajayant Misra (M'09–SM'05) received the M.Sc. degree in physics and information systems from BITS, Pilani, India, in 2003, and the Ph.D. degree in computer science from Arizona State University, Tempe, AZ, USA, in 2009.

He is currently an Associate Professor of computer science with New Mexico State University, Las Cruces, NM, USA. His research interests include wireless networks and the Internet, supercomputing, and smart grid architectures and protocols.

Dr. Misra has served on several Editorial Boards of IEEE publications and conference Executive Committees (*Communications on Surveys and Tutorials, Wireless Communications Magazine*, SECON 2010, INFOCOM 2012). He has authored more than 50 peer-reviewed IEEE/ACM journal papers and conference proceedings, which have garnered over 3000 citations. More information can be obtained online at www.cs.nmsu.edu/~misra.