The Missing Piece: On Namespace Management in NDN and How DNSSEC Might Help

Pouyan Fotouhi Tehrani
Weizenbaum Institute /
Fraunhofer FOKUS
pft@acm.org

Eric Osterweil
George Mason University
eoster@gmu.edu

Jochen H. Schiller
Freie Universität Berlin
jochen.schiller@fu-berlin.de

Thomas C. Schmidt
HAW Hamburg
t.schmidt@haw-hamburg.de

Matthias Wählisch
Freie Universität Berlin
m.waehlisch@fu-berlin.de

ABSTRACT

Names are the cornerstone of every *Information-Centric Network* (ICN), nonetheless, namespace management has been by far neglected in ICN. A global and scalable namespace management approach is a challenge which not only concerns technical, but also requires attention to non-technical, *e.g.,* organizational issues. In this paper, we present both a clear position on namespace management in ICN and preliminary work on a potential solution. We conceptualize a namespace management system for hierarchical names and introduce a prototype for NDN, which leverages existing DNSSEC equipped DNS infrastructure. Based on this, we are able to implement both technical and non-technical aspects of namespace management. We consider lessons learned and pitfalls from decades of the ever-evolving development of domain name system. As the de facto standard namespace management for the Internet, it is an integral orientation factor for both our concept and its implementation.

CCS CONCEPTS

• **Networks → Naming and addressing**; **Naming and addressing**; *Application layer protocols*;

KEYWORDS

Namespace Management, Data Origin Authentication, ICN, NDN, DNS, DNSSEC

ACM Reference Format:
Pouyan Fotouhi Tehrani, Eric Osterweil, Jochen H. Schiller, Thomas C. Schmidt, and Matthias Wählisch. 2019. The Missing Piece: On Namespace Management in NDN and How DNSSEC Might Help. In *6th ACM Conference on Information-Centric Networking (ICN '19), September 24–26, 2019, Macao, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3357150.3357401

ICN '19, September 24–26, 2019, Macao, China

ACM ISBN 978-1-4503-6970-1/19/09…$15.00
https://doi.org/10.1145/3357150.3357401

1 INTRODUCTION

Names and their management have been considered crucial, right from the beginning of the Internet and its predecessor [23, 24, 33, 34]. Originally, names were administered by John Postel, later, to scale with the growth of the Internet, the *Internet Corporation for Assigned Names and Numbers* (ICANN) was designated for organizational aspects, such as name assignment, policy development and enforcement, and dispute mediation. This enabled the Domain Name System [25] (DNS) to become fundamental for almost all Internet applications. It not only defines a technical solution to resolve hierarchically structured domain names (in the technical protocol) but also provides an ecosystem that ensures proper name management (in the policy organization and community).

As the Internet's most relied upon technology that facilitates namespace management, which matured from research-labs into operational and commercial environments and has gone on to implement real-world multi-stakeholder deployment, the long-established DNS ecosystem is the canonical example that illustrates a clear need for some kind of namespace management in computer networks. In the Information-Centric Networking (ICN) community and literature, which introduces names as first class principles, this topic remains largely ignored. Current work on naming security in ICN mainly focuses on securing name to data bindings [10, 13, 16, 43] and provenance authentication [2, 10, 11, 16, 36, 39, 43]. Existing solutions are respectively limited to technical solutions while overlooking the derivative (but separate) need to attend the policy and organizational aspects.

In this paper, we argue for two positions: (*i*) A successful ICN deployment-model requires clarification of namespace policy and management, and (*ii*) augmenting the design of an ICN solution with the long-term facilities and experiences gained from the DNS ecosystem would allow us to leverage existing structures to address and overcome the significant number of technical, legal, and policy challenges that are inevitable as ICNs achieve operational and industry deployments.

In detail, we revisit the namespace management problem (**§ 2**) and present preliminary work to tackle the inevitable (often not only technical) challenges NDN will face in bolstering operational ICN deployment. To reach acceptance among multiple operational and commercial stakeholders, ICN needs to guarantee (*i*) exclusive ownership of names by publishers, (*ii*) the arbitration of rightful holders of named resources, and (*iii*) the verification of the content

origin by consumers. To achieve this, we conceptualize (**§ 3**) and implement a scalable namespace management scheme while recognizing the fact that namespace management is only effective if both technical and organization aspects are attended [19]. We show the feasibility of our approach by presenting NDNSSEC (**§ 4**), a prototype that incorporates *DNS security extensions* (DNSSEC) [28] attestation objects into NDN to prove name ownership.

2 NAMESPACE MANAGEMENT

A namespace $\mathcal{N}$ denotes a "set of names from which all names for a given collection of objects are taken" [9, §8]. A name is a unique string in an alphabet referring to some object; the set of all objects to which a name might be *bound* is called the *scope* of that namespace. A namespace, thus, can be formalized as a mapping, *i.e.,* a *functional binary relation*, with $\mathcal{N}$ and $\mathcal{T}$ being respectively its domain and codomain. Such mapping is functional but neither surjective, so that a thing can be left without a name assignment, nor injective to cater for name aliases. In the following, the notation $\mathcal{T}$, standing for set of *things*, is used to denote the scope of a namespace, and $\mathcal{N} \rightarrow \mathcal{T}$ denotes the binding of names to things.

Given a namespace $\mathcal{N}$, a (decentralized) namespace management scheme partitions $\mathcal{N}$ into management units, *zones* [12, §6], which are owned and maintained by an authoritative entity. Namespace management binds a name $n \in \mathcal{N}$ to a zone $\mathcal{Z}_i \in \mathcal{Z}$ where $\mathcal{Z}$ denotes the set of possible zones with each zone $\mathcal{Z}_i$ comprising a number of names. The zone owner manages or delegates the management of all or a subset of names within that zone. We use the notation $\mathcal{N} \rightarrow \mathcal{Z}$ to denote binding of names to zones, which similar to $\mathcal{N} \rightarrow \mathcal{T}$ is functional and left-total so that each name is mapped exactly to a single zone. It is neither necessarily surjective, so that there might be zones which include no names, *i.e.,* are not assigned, nor injective to allow mapping of multiple names to the same zone. Needless to say, manageability is proportional to the number of bound names; using zones reduces management overhead, while preserving the overall size of the namespace and improving the scalability.

Regardless of a formal definition and respective technical means of implementation and deployment, we argue that an effective global namespace management scheme, specifically for computer networks, must also address non-technical issues: not only name ownership must be guaranteed and protected but the management scheme itself must conform to constraints beyond the technical realm. An organization or a company might want to continue using terms, such as trademarks or any other distinguishable and established names associated with them in the real world, in order to remain recognizable in the virtual world. From a purely technical point, such associations are irrelevant and out of the scope of namespace mangement: a point of view, which we consider as short-sighted. In the following, we support our claim by revisiting the development history of DNS in the past decades.

The domain name system (DNS), as the most prominent example of namespace management in computer networking, is a hierarchical distributed key-value database, with $\mathcal{N}$ being the set of all possible domain names, $\mathcal{T}$ being Internet resources, and $\mathcal{Z}$ the set of *DNS zones*. When centralized name management became infeasible, DNS was conceived. The centralized administration was divided into sub-administrations [27], organized hierarchically in a tree structure, and meant to mirror the organizational structure of its managing authority [24]. Domain names were initially conceptualized as purely technical administrative entities allocated on a first-come / first-served basis. They were required only to be registered with the *central domain administrator*, have a designated maintainer, and provide their own name lookup service [27]. As the Internet began its commercializing phase and grew in terms of participating nodes, assigned domains, and users, it became evident that in reality a domain name goes beyond the purely technical context. In 1996, for example, a district court in Cologne, Germany, decided that a domain name corresponding to a city name is *not* subsumed under the naming laws of the German Civil Law. It was argued that a domain name is comparable with "telephone numbers, bank routing numbers or postal codes" and does not necessarily "establish an association between the domain name and its owner" [21]. In less than a year, another district court, this time in Frankfurt, Germany, decided the contrary. It argued that users not only expect to retrieve information *about* but also *from* the municipality of a city under the corresponding domain name while rejecting the previous comparison with phone numbers [20]. It was decided that the domain namespace is not limited to the former interpretation, *i.e., where to send the data* [9], but also regards the latter context, *i.e., indicating the origin* [19] in terms of the real-world owner of that name, and involves naming and trademark laws.

Analogous conflicts have arisen in the overlapping uses of domain names by independent resolution systems (global DNS versus internal DNS namespaces, for example). In cases where names appear similar but have conflicting meanings in different namespaces, name collisions have led to security vulnerabilities [5]. Such disputes and ambiguities (and others) were among the incentives (and types of incentives) for establishment of ICANN with the goal of not only addressing technical, but also non-technical manageability of DNS [26]. The reason is that different *contexts*, related to which a name is interpreted [30], are not always clearly separated or are even intentionally overlapping. It is, thus, possible for a name or its equivalent variations to be *resolved* to different things in the *context* of different namespaces. The correspondence of different namespaces might cause unanticipated complications that need to be addressed non-technically. Such correspondence is not necessarily self-evident and needs to be acknowledged or established formally.

Finally, there are other less obvious yet non-trivial concerns regarding global namespace management that highlight the importance of its non-technical aspects. For example, depending on the scope of a namespace, having authority over a zone can asymmetrically put the owner in a position of power. Considering a hierarchically structured namespace where each zone apex denotes a subject, *e.g.,* '`/news/world/politics`', the owner of top-level apexes can influence the discourse of each subject, for example, by limiting the delegation only to favorable news outlets and suppressing the rest.

3 CONCEPTUAL DESIGN

The $\mathcal{N} \rightarrow \mathcal{T}$ binding is the basis of any ICN with $\mathcal{N}$ being the ICN's namespace defined through a naming scheme and $\mathcal{T}$ the data

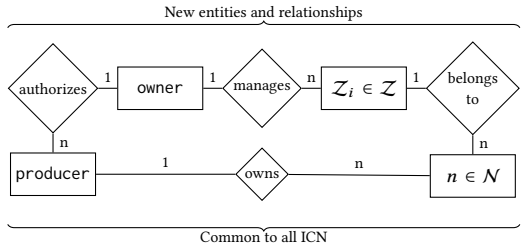

Figure 1: Entity-relationship diagram for a simple name space management framework.

objects within that ICN. Nevertheless, namespace management has not adequately been addressed in ICN. On the one hand, there exists no single authority which is in charge of global namespace allocation and assignment. On the other hand, there is no consensus on how to define ownership/membership of names or amalgomation into zones.

The first step in realizing namespace management in ICN is to acknowledge that the named data is served by different data holders and to codify the inherent subdivision of the namespace into management units, *i.e.,* zones. For the sake of simplicity we assume that zones are non-overlapping and a name only belongs to a single management unit so that the following holds $\forall i, j : \mathcal{Z}_i \cap \mathcal{Z}_j = \emptyset$. Names are then to be mapped to a zone, establishing a functional, $\mathcal{N} \rightarrow \mathcal{Z}$ mapping. Here we limit ourselves to structured naming schemes which allow embedding of zone identifiers in names, *e.g.,* as prefixes in hierarchical naming schemes or as authority field of URI-based schemes. For example, both '`ni://`**`ietf.org`**`/sha-256;Uya. . . _-Q`' (*cf.,* [13]) and '**`/org/ietf`**`/index.html`' contain zone apexes (in bold) of a hierarchically structured namespace directly in the name.

Furthermore, a logical authoritative entity is designated to assign and allocate zones to owners, *i.e.,* mapping zones to owners. A zone owner can then delegate management, authorize producers to publish under names belonging to that zone, and provide information about authorized producers. The relationship between zones, owners, producers, and names, as required for namespace management in ICN, is depicted in Figure 1.

Finally, given a binding, it should be possible to authenticate the binding, *i.e.,* to verify its validity, within the context of its respective namespace. In the following, we discuss two methods of securing bindings: the first method allows to authenticate the mapping by its intrinsic properties, whereas the second one relies on third parties to vouch for its correctness and validity.

Self-authentication/certification. A binding is said to be self-authenticating if its verification succeeds only by the bound values and locally available information. For example, if $\mathcal{N} \rightarrow \mathcal{T}$ is defined as $n = h(t)$ with t being a binary object and h a secure cryptographic hash function, the authenticity of the mapping between the name n and object t can easily be verified by calculating hash of t using h and comparing it with n, given that the applied hash algorithm is known. *Self-certification* requires the additional knowledge of a cryptographic key $k \in \mathcal{K}$ to authenticate the binding [15]. The main drawback of such bindings for names is that they diminish

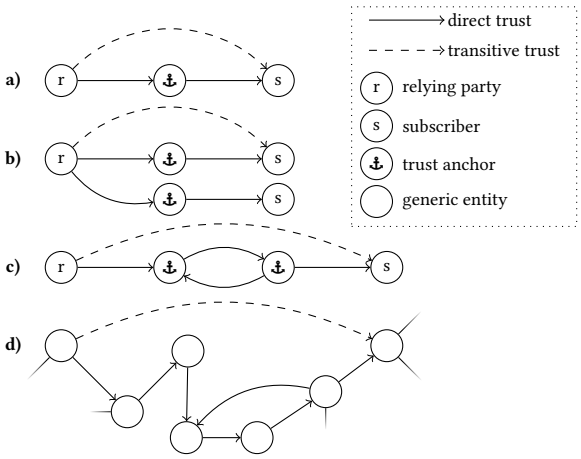

Figure 2: Trust models—a) Basic Trust Model, b) Multiple Trust Anchors, c) Cross Certified Trust Anchors, d) Web of Trust.

readability for humans and prevent aggregation, *e.g.,* in forwarding tables.

Trusted Third Parties (TTP). Beside self-authentication or certification, trusted third parties can be designated to directly vouch for and verify a given binding within a *trust management* framework. Involved parties are identified by their digital credentials, *i.e.,* cryptographic keys $\in \mathcal{K}$, and their actions are constrained by defined policies, while a *trust relation* denotes which authorities may issue credentials [3]. Trust relations are realized through certificates and trust can be established transitively when a *trust anchor* delegates credential issuance to a *subscriber* which subscribes to services of the TTP [35, p. 937]. Trust between a *relying party r* and a subscriber *s* is then transitively established through the TTP as depicted in Figure 2a. Alternative models may contain multiple trust anchors (Figure 2b) or cross certified trust anchors (Figure 2c). Prominent examples are the single rooted DNSSEC and multiple (cross certified) trust anchors model of Web PKI based on X.509 [4] certificates. A generalization of this approach is the Web of Trust (WOT) where every entity acts both as an authority and a relying party. Trust between two parties is then considered to be established if there exists a trust relation path between them [44] (Figure 2d).

In our concept, we define a simple policy for data publishing under a given zone: a producer is authorized to publish under that zone, if its public key is certified by the zone owner. The producer, in turn, must sign data before publication with the corresponding private key. Respectively, a consumer can verify if the publisher was authorized by the zone owner or not.

4 IMPLEMENTATION OF NDNSSEC

In our implementation, NDNSSEC, we couple NDN as our ICN core with the existing DNS and DNSSEC infrastructure as the name managing unit. Our prototype extends present NDN tools to allow (*i*) NDN producers to publish and sign data and (*ii*) consumers to verify the data, based on existing DNSSEC data and with almost

no manual interaction as long as name ownership is secured via DNSSEC.

DNS has been chosen as it not only adequately addresses technical but also non-technical challenges of a scalable global namespace management—and it is deployed, while NDN has the benefit of a hierarchical naming scheme that matches the hierarchical structure of DNS namespace. It is noteworthy that NDN explicitly excludes namespace management from its architectural design [6] and leaves it for the application layer. Our dependence on DNS is limited to its ecosystem and publicly available data and not necessarily its transport.

The software of our working prototype[1] is based upon *NDN C++ library with eXperimental eXtensions* (ndn-cxx) version 0.6.5 [7].

4.1 Securing Bindings

Namespace management requires four bindings (see Figure 1). We implement two of them based on existing built-in data-oriented security mechanisms of NDN, and the other two by integrating the attestation objects of DNSSEC.

Named data packets in NDN carry a name, payload, and metadata including signing information. A packet can be secured either using a hash digest for integrity examination or by a digital signature catering for both integrity and authenticity verification. For digital signatures, the signature block of an NDN data packet includes a `KeyLocator` field which denotes under which name the certificate of the signing party can be retrieved. Certificates in NDN are ordinary data packets which carry signed public keys as payloads.

In the following, the terms *name* and *producer* respectively denote the packet identifier and the entity responsible for its creation and provision [38]. We use the term *zone apex* to denote a DNS zone origin, *i.e.,* the name at the root of a zone tree [17, §7]. A zone apex is either a conventional fully qualified domain name (FQDN) or its equivalent *reverse slash separated* notation, *e.g.,* '`tools.ietf.org.`' or '`/org/ietf/tools`' (*cf., ndnification* [1, §3]). *Zone owners*, are real-world entities which are authorized to manage the namespace within a zone.

In our selected trust management framework based on DNSSEC, ICN nodes, *e.g.,* producers and consumers, which can be subject to authentication are represented through their cryptographic keys or more specifically through NDN certificates. There exists a (logically) singular trust anchor, the root zone owner, and the remaining zone owners act as its subscribers. Similar to ICN nodes, zone owners, are also represented through cryptographic keys listed as `DNSKEY`s in a zone's authoritative data. Securing bindings succeeds as follows:

Name to Zone. We establish name to zone bindings by prefixing names with zone apexes. As zone apexes are embedded into names, $\mathcal{N} \rightarrow \mathcal{Z}$ mappings are transitively authenticated by authenticating name to producer and producer to zone bindings.

Name to Producer. This binding is already given in NDN. A data packet binds its names to a producer through a digital signature [42].

Producer to Zone. This binding is realized using `DNSKEY` and `RRSIG` records of DNSSEC. The zone owner enlists the public key of producers as `DNSKEY`s to indicate producers which are authorized to publish under its managed zone.

[1]Source code available under `https://gitlab.com/ndnssec`

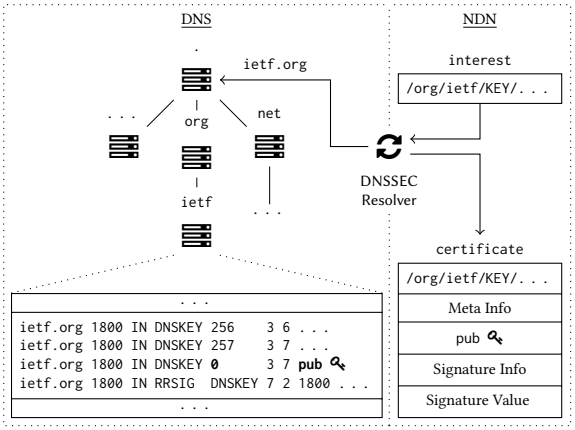

Figure 3: Resolver as bridge between DNS and NDN.

Zone to Zone Owner. DNS zone delegation is leveraged to bind zones to zone owners.

4.2 Workflow

Prior to publishing under a zone apex, a producer is required to have its public key served by the DNS zone owner as a `DNSKEY` record. Upon publication, the `KeyLocator` field of the NDN packet is set to `/<SubjectName>/KEY/[KeyId]/[IssuerId]/[Ver]` (see [40] for certificate naming convention in NDN), with `SubjectName` denoting the zone apex, and `KeyId` the digest of the producer's public key calculated the same way digests are produced for DNS `DS` records [29, §5].

Upon receiving a data packet, our extended NDN consumer software first verifies that the content name is prefixed with a zone apex matching the `SubjectName` segment of its `KeyLocator`. Otherwise, the packet is discarded. Then, the `DNSKEY`s under the zone apex are fetched via DNS, their digests are computed, and finally compared with the `KeyId` to find the matching public key. If a match is found, it can be used to verify the signature of the packet. It is also possible for each domain to provide a logical public key resolver (similar to a DNS stub resolver) which is trusted by consumers of that domain: given a zone apex and a public key digest, the resolver verifies whether the key is served by the zone owner as authorized and returns a corresponding public key for that producer as depicted in Figure 3, otherwise it returns an explicit `NACK`. Moreover, a domain-local resolver enables *private zone* management services for zone apexes which are explicitly used for internal purposes, similar to *internal top-level domains* (iTLD).

Although our prototype relies on the DNS transport to fetch zone information, there are alternative methods to integrate publicly available DNS data into NDN without compromising the integrity of our approach. A zone owner can, for example, publish an *authentication chain* (*cf.,* [32]) as a simple NDN packet which comprises signed DNS record sets all the way from a trust anchor to the zone it manages so that consumers can fetch and authenticate zone records without having to consult the respective name servers. An alternative is to mirror DNS data using NDN-specific distributed key-value databases resembling the DNS approach, *e.g.,* NDNS [1, 2]. Either

Table 1: Complexity of certificate chain verification.

	Level 1 (root)	Level i (interim)	Level n (leaf)	$\lvert\text{⚓}\rvert$
Trust Schema	⚓ C_0 ⟵ ⋯ — C_i ⟵ ⋯ — C_n			≥ 1
NDNS	⚓ KSK_0 ↘ DSK_0	↗ $DKEY_i$ ↘ KSK_i ↘ DSK_i	↗ $DKEY_n$ ↘ KSK_n ↘ DSK_n ↘ $APPCERT_n$	≥ 1
NDNSSEC	⚓ $DNSKEY_0$ ↘ NS_0 ↘ DS_0	↗ $DNSKEY_i$ ↘ NS_i ↘ DS_i	→ $DNSKEY_n$	$= 1$

approach, however, introduces a synchronization problem which can lead to temporary discrepancies between DNS data and their NDN representation. A solution for this is beyond the scope of this work.

4.3 Evaluation

Performance evaluation of our proposed approach mainly reduces to evaluating DNS, which has been extensively researched elsewhere and is out of the scope of this work. A comparative evaluation is also not meaningful due to lack of existing alternatives. Instead, we focus on a qualitative analysis of data authentication and the computational complexity of our trust management model by comparing it to NDN's schematic trust [41] and NDNS [1, 2] as alternative trust management frameworks.

The trust policy of DNSSEC can be realized using trust schemata by defining a single trust rule as follows [41]:

Rule	Data Name	Key Name
`cert`	`(<>*)(<>)<KEY>[id][iid][v]`	`cert(\1,null) | root()`

This rule states that for any given certificate, the certificate of its signing entity can be found under the same name but on a level higher in zone hierarchy until a trust anchor (`root()`) is reached. A consumer starts by fetching certificates for each zone, starting at level n, up to a known trust anchor and verifies the certificate chain back to the zone certificate. NDNS proposes an approach resembling that of DNSSEC: the chain verification starts at a known trust anchor and ends with a certificate. Both approaches allow multiple trust anchors to coexist with a high price of i) complicating the trust bootstrapping phase, ii) requiring consumers to redo chain verification, if selected trust anchor does not lead to the certificate under investigation, and iii) increasing management overhead needed to avoid policy collisions. Table 1 depicts how each approach traverses from a trust anchor to a certificate used to sign a data packet.

Furthermore, NDNSSEC proposes a different method to address certification validity verification and revocation which directly determines the certificate retrieval frequency by a consumer, *e.g.,* when authenticating data streams. Whereas in NDN certificates carry a *freshness period*, comparable to DNS TTL, alongside a *validity period* (also common in X.509 certificates), in our approach, we only use TTL and leave it to consumers to decide on an appropriate policy regarding the validity period of retrieved public keys, *e.g.,* for caching. In NDNSSEC, a public key listed as a `DNSKEY` is considered to be valid and no further revocation mechanisms are required. It should, however, be noted that DNS updates are constrained by propagation delays [14] so there might be short periods of time when retrieved record sets are out of sync.

Comparing to aforementioned approaches, advantages of NDNSSEC can be summarized as follows:

(1) **Non-technical policy enforcement** by relying on existing organizations such as ICANN.
(2) **Minimal management burden on consumers** by designating a single party as trust anchor (DNS root).
(3) **Deterministic authentication** by always guaranteeing a path from the root to a zone owner, otherwise signaling that a zone has not been assigned or delegated through *explicit denial of existence* [37] of DNSSEC .
(4) **No additional infrastructure for certificate revocation** by simply replacing compromised or outdated keys through new `DNSKEY` records (cf. *suicide lists* in NDN [40]).

5 RELATED WORK

The authors are not aware of any previous work that has elaborated the non-technical policy aspects of namespace management in ICN. In this section, the review of related work is, thus, limited to technical issues, such as authenticating bindings and establishing trust.

Self-certifying names. Self-authentication/certification has been proposed in a number of approaches to secure $\mathcal{N} \rightarrow \mathcal{T}$ relations. In CONET [18], names are structured as $P : L$ where P denotes the public key's digest of a producer and L the cryptographic hash of the label [10]. Similarly, in MobilityFirst [36], principal names are self-certified, *i.e.,* are a digest of the owner's public key, whereas names of data objects are cryptographic hashes of the data itself, *i.e.,* are self-authenticating. Using hashes as names is also proposed in *NetInf* [8] within a URI-based naming scheme where the path component represents the base64 URL encoded digest of the content [13].

Trusted Third Parties. Wong and Nikander [39] leverage a URI-based naming scheme with producer's identity placed as authority segment. A centralized resolution server resolves identities to a public key while mapping the resource path of the URI to the object's metadata, including its digital signature. A distributed solution of the problem provides *NDNS*, an always-on distributed lookup service for NDN [1, 2]. Inspired by DNS, NDNS maps names to a number of records used as routing hints (`NS`), certificates (`APPCERT`, `CERT`), and general records (`TXT`) [2]. Here, the managed namespace is a subset of NDN namespace, which can bidirectionally be translated into domain names. Similar to NDNS, *key resolution service* (KRS) [22] introduces a distributed resolution mechanism, but for CCN. It can resolve a name into at least a content hash, publisher certificate, or a certificate chain. Both in NDNS and KRS, authenticating $\mathcal{N} \rightarrow \mathcal{T}$ and $\mathcal{N} \rightarrow \mathcal{Z}$ relations succeeds indirectly through publisher authentication. DiBenedetto and Papadopoulos [11] propose to use a resolution service, such as KRS or NDNS, to map a name to the public key of the domain owner. The corresponding private key acts as a key-signing key (KSK) so that any producer with its key certified by the zone KSK is considered as authorized to publish under the corresponding zone.

Identity-based Cryptography (IBC). To prevent storing public keys at third parties, IBC [31] has been proposed. In IBC an identity,

Table 2: Namespace and trust management in ICN with $\mathcal{N}$ denotes names, $\mathcal{T}$ things, $\mathcal{Z}$ zones, $\mathcal{I}$ identities.

	Authentication					
	$\mathcal{N} \to \mathcal{T}$	$\mathcal{N} \to \mathcal{Z}$	Self-auth	Zone Scope	Trust Relation	Naming Example
DNSSEC [28]	✓	✓	✘	$\mathcal{N}$	Decentralized	`tools.ietf.org`
Detti et al. [10]	✓	✓	✓*	$\mathcal{I}$	Decentralized	`<a0914a9. . . 7287a, 0902908. . . 6ec006a>`
Venkataramani et al. [36]	✓	✘	✓*	$\emptyset$	N/A	`8c6a365205c874144. . . 3765a2190404de6a9`
Farrell et al. [13]	✓	✘	✓	$\emptyset$	N/A	`ni:///sha-256;UyaQV-Ev. . . 1aGQAlMO2X_-Q`
Wong and Nikander [39]	✓	✓	✘	$\mathcal{I}$	Centralized	`scheme://authority/page.html`
Afanasyev [1, 2]	✓	✓	✘	$\mathcal{N}$	Decentralized	$/\text{level}_1/\ldots/\text{level}_n$`/data/name`
Mahadevan et al. [22]	✓	✓	✘	$\mathcal{N}$	Decentralized	$/\text{level}_1/\ldots/\text{level}_n$`/data/name`
DiBenedetto and Papadopoulos [11]	✓	✓	✘	$\mathcal{N}$	Decentralized	$/\text{level}_1/\ldots/\text{level}_n$`/data/name`
Zhang et al. [43]	✓	✓	✓*	$\mathcal{I}/\mathcal{N}$	Decentralized	`/arbitrary/ndn/conform/name`
Hamdane et al. [16]	✓	✓	✓*	$\mathcal{I}$	Decentralized	`/producerID/contentID/validity/ver/seg#`

* Denotes *self-certification* (see Section 2)

e.g., content name, is used to generate the corresponding public key using parameters provided by a trusted *private key generator* (PKG). Zhang et al. [43] propose an IBC based method where a CCN/NDN compatible hierarchical name $n \in \mathcal{N}$ is used as identity. A producer can also prefix a name with its identity $i \in \mathcal{I}$ to denote ownership. Data packets here carry metadata including the producer's identity, establishing a $\mathcal{N} \to \mathcal{I}$ relation, and the respective PKG public parameters, enabling self-certification of $\mathcal{N} \to \mathcal{T}$ relations. If names are prefixed with producer IDs, the producer fetches the corresponding key-pair directly from PKG, otherwise a name resolution server (NRS), responsible for key management and policy enforcement, acts as a proxy to the PKG. In either case, consumers only need to fetch the public parameters of the PKG to authenticate data packets. The authors also propose a hybrid solution with domain level PKG/NRS managing *domains* alongside a global PKI for their hierarchical management.

A variation of IBC, namely hierarchical IBC (HIBC), where key generation and identity authentication is delegated among hierarchical PKGs, is leveraged by Hamdane et al. to enhance NDN with a custom naming structure to enable producer identification and authentication alongside integrity verification [16]. In the proposed scheme each segment of an NDO's name n is mapped to a designated PKG which implicitly defines a $\mathcal{N} \to \mathcal{I}$ relation where $i \in \mathcal{I}$ denotes an organizational entity. Here, having the parameters of the root PKG allows the authentication of credentials issued throughout the hierarchy but suffers from the *key-escrow* problem, *i.e.,* a PKG has or can generate private key of its children.

Comparison. We summarize our observations in Table 2. All approaches cater for authenticating $\mathcal{N} \to \mathcal{T}$ and mostly allow for defining zones as collection of names by the same producer or as an independent entity. Self-authentication, either for $\mathcal{N} \to \mathcal{T}$, $\mathcal{N} \to \mathcal{Z}$, or both, has been leveraged alongside centralized or decentralized trusted third parties where self-authentication does not suffice. To authenticate TTPs, *e.g.,* resolution servers and PKGs, all approaches use some variation of chain of trust either with domain-local or global trust anchors. Regardless of which method is used to establish trust, a *trust bootstrapping* phase is necessary to establish trust anchors or trusted credentials in general. In contrast to this paper, all methods assume this phase as given without giving further details.

6 CONCLUSION AND RESEARCH ROADMAP

The *technical* mechanics of a namespace are critical to ICNs, but there are many aspects to operational networks and ecosystems that extend beyond just the technical landscape. With over 30 years of experience in global namespace management of over 354 million domains and billions of euros in business per year, the DNS industry has illustrated that policy and management aspects of namespace resolution are critical components. The authors believe that this is the first work to address the *necessary* requirements of namespace policy and management for ICNs, and this is presented through the lens of using the resources that exist in the operational Internet, today. This work underscores the observation that names are not just labels used to identify things, they require policy and context. It is, thus, important to manage names and how they are used. It is also crucial for users to be able to examine if a name is used by its authorized owner, and have policy frameworks to codify "authorization."

In addition to formalizing namespace management for ICNs, we have proposed a concrete solution for hierarchically structuring and assigning the namespace of NDN, based on DNSSEC. We have shown how minimal changes to existing mechanisms of NDN can enable collision-free and scalable namespace management, which addresses both technical and non-technical shortcomings. Finally, as the success of the Internet is driven by community efforts, we propose the following research roadmap. In the short term, integrating DNS data into the ICN ecosystem without relying on DNS-specific transport is an important step. Such activities should consider the performance impact on the ICN ecosystem in terms of synchronization disparities, which might ensue and affect various aspects of publishing, discovery, and retrieval of data objects. In the long run, there is clearly value in the evaluation of our proposed approach with respect to large-scale user studies, *e.g.,* by modeling real-world settings based on traces from global cloud providers. Upcoming work should also probe the feasibility of NDNSSEC in confined use cases such as disaster scenarios with intermittent connectivity or fragmented networks. Making the right trustworthy public keys locally available without access to the public key infrastructure is challenging but might benefit from ICN in-network caching.

Acknowledgments. We would like to thank the anonymous reviewers and our shepherd Karen Sollins for their concise and constructive feedback. We appreciate the help of Luca Keidel, who provided the first version of our prototype. This work was supported in parts by the German Federal Ministry of Education and Research (BMBF) within the projects *I3* and *Deutsches Internet-Institut* (grant no. *16DII111*).

REFERENCES

[1] Alexander Afanasyev. 2013. *Addressing Operational Challenges in Named Data Networking Through NDNS Distributed Database.* Ph.D. Dissertation. University of California Los Angeles.

[2] Alexander Afanasyev, Xiaoke Jiang, Yingdi Yu, Jiewen Tan, Yumin Xia, Allison Mankin, and Lixia Zhang. 2017. NDNS: A DNS-Like Name Service for NDN. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 1–9.

[3] Matt Blaze, Joan Feigenbaum, and Jack Lacy. 1996. Decentralized trust management. In *Proceedings 1996 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 164–173.

[4] Sharon Boeyen, Stefan Santesson, Tim Polk, Russ Housley, Stephen Farrell, and Dave Cooper. 2008. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280.

[5] Qi Alfred Chen, Matthew Thomas, Eric Osterweil, Yulong Cao, Jie You, and Z. Morley Mao. 2017. Client-side Name Collision Vulnerability in the New gTLD Era: A Systematic Study. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 941–956.

[6] NDN Consortium. 2019. Named Data Networking: Motivation & Details. `https://web.archive.org/web/20190408131300/https://named-data.net/project/archoverview/`

[7] NDN Consortium. 2019. ndn-cxx: NDN C++ library with eXperimental eXtensions 0.6.5. `https://web.archive.org/web/20190325105322/http://named-data.net/doc/ndn-cxx/0.6.5/release-notes/release-notes-0.6.5.html`

[8] Christian Dannewitz, Jovan Golic, Börje Ohlman, and Bengt Ahlgren. 2010. Secure Naming for a Network of Information. In *2010 INFOCOM IEEE Conference on Computer Communications Workshops*. IEEE, 1–6.

[9] J. Day. 2007. *Patterns in Network Architecture: A Return to Fundamentals*. Pearson Education.

[10] Andrea Detti, Nicola Blefari Melazzi, Stefano Salsano, and Matteo Pomposini. 2011. CONET. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking - ICN '11*. ACM Press, New York, New York, USA, 50–55.

[11] Stephanie DiBenedetto and Christos Papadopoulos. 2016. Mitigating poisoned content with forwarding strategy. In *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 164–169.

[12] Robert Elz and Randy Bush. 1997. Clarifications to the DNS Specification. RFC 2181.

[13] Stephen Farrell, Dirk Kutscher, Christian Dannewitz, Börje Ohlman, Ari Keränen, and Phillip Hallam-Baker. 2013. Naming Things with Hashes. RFC 6920.

[14] Z. Gao and A. Venkataramani. 2019. Measuring Update Performance and Consistency Anomalies in Managed DNS Services. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. 2206–2214.

[15] Ali Ghodsi, Teemu Koponen, Jarno Rajahalme, Pasi Sarolahti, and Scott Shenker. 2011. Naming in content-oriented architectures. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking - ICN '11*. ACM Press, New York, New York, USA, 1–6.

[16] Balkis Hamdane, Rihab Boussada, Mohamed Elhoucine Elhdhili, and Sihem Guemara El Fatmi. 2017. Hierarchical Identity Based Cryptography for Security and Trust in Named Data Networking. In *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE, 226–231.

[17] Paul E. Hoffman, Andrew Sullivan, and Kazunori Fujiwara. 2019. DNS Terminology. RFC 8499.

[18] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. 2007. A data-oriented (and beyond) network architecture. *ACM SIGCOMM Computer Communication Review* 37, 4 (10 2007), 181–192.

[19] Marshall Leaffer. 1998. Domain Names, Globalization, and Internet Governance. *Indiana Journal of Global Legal Studies* 6, 1 (1998), 139–165.

[20] LG Ansbach. 1997. 2 O 99/97 – Verletzung des Namensrechts durch Internet-Adresse – ansbach.de.

[21] LG Köln. 1996. 3 O 507-96 – Verwendung von Domain-Namen – Pulheim.de.

[22] Priya Mahadevan, Ersin Uzun, Spencer Sevilla, and J.J. Garcia-Luna-Aceves. 2014. CCN-KRS. In *Proceedings of the 1st international conference on Information-centric networking - INC '14*, Vol. 94304. ACM Press, New York, New York, USA, 97–106.

[23] D. L. Mills. 1981. Internet name domains. RFC 799.

[24] P. Mockapetris. 1983. Domain names: Concepts and facilities. RFC 882.

[25] P. Mockapetris and K. J. Dunlap. 1988. Development of the domain name system. *ACM SIGCOMM Computer Communication Review* 18, 4 (08 1988), 123–133.

[26] National Telecommunications and Information Administration. 1998. Improvement of Technical Management of Internet Names and Addresses. *Federal Register* 63, 34 (20 02 1998), 8826–8833.

[27] J. Postel and J. Reynolds. 1984. Domain requirements. RFC 920.

[28] Scott Rose, Matt Larson, Dan Massey, Rob Austein, and Roy Arends. 2005. DNS Security Introduction and Requirements. RFC 4033.

[29] Scott Rose, Matt Larson, Dan Massey, Rob Austein, and Roy Arends. 2005. Resource Records for the DNS Security Extensions. RFC 4034.

[30] J. H. Saltzer. 1978. Naming and Binding of Objects. In *Operating Systems - An Advanced Course*. Springer, Berlin, Heidelberg, 99–208.

[31] Adi Shamir. 1985. Identity-based Cryptosystems and Signature Schemes. In *Proceedings of CRYPTO 84 on Advances in Cryptology*. Springer-Verlag New York, Inc., New York, NY, USA, 47–53.

[32] Melinda Shore, Richard Barnes, Shumon Huque, and Willem Toorop. 2018. *A DANE Record and DNSSEC Authentication Chain Extension for TLS*. Internet-Draft draft-ietf-tls-dnssec-chain-extension-07. Internet Engineering Task Force. Work in Progress.

[33] Z. Su. 1982. Distributed system for Internet name service. RFC 830.

[34] Z. Su and J. Postel. 1982. The Domain Naming Convention for Internet User Applications. RFC 819.

[35] H.C.A. van Tilborg and S. Jajodia. 2011. *Encyclopedia of Cryptography and Security*. Springer US.

[36] Arun Venkataramani, James F. Kurose, Dipankar Raychaudhuri, Kiran Nagaraja, Morley Mao, and Suman Banerjee. 2014. MobilityFirst: A Mobility-centric and Trustworthy Internet Architecture. *SIGCOMM Comput. Commun. Rev.* 44, 3 (07 2014), 74–80.

[37] Samuel Weiler and Johan Ihren. 2006. Minimally Covering NSEC Records and DNSSEC On-line Signing. RFC 4470.

[38] Bastiaan Wissingh, Christopher A. Wood, Alex Afanasyev, Lixia Zhang, David R. Oran, and Christian Tschudin. 2019. *Information-Centric Networking (ICN): CCN and NDN Terminology*. Internet-Draft draft-irtf-icnrg-terminology-04. Internet Engineering Task Force. Work in Progress.

[39] Walter Wong and Pekka Nikander. 2010. Secure naming in information-centric networks. In *Proceedings of the Re-Architecting the Internet Workshop on - ReARCH '10*. ACM Press, New York, New York, USA, 12:1–12:6.

[40] Yingdi Yu. 2015. *Public Key Management in Named Data Networking*. Technical Report. UCLA. 1–8 pages.

[41] Yingdi Yu, Alexander Afanasyev, David Clark, Kc Claffy, Van Jacobson, and Lixia Zhang. 2015. Schematizing Trust in Named Data Networking. In *Proceedings of the 2nd International Conference on Information-Centric Networking - ICN '15*. ACM Press, New York, New York, USA, 177–186.

[42] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Kc Claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. 2014. Named data networking. *ACM SIGCOMM Computer Communication Review* 44, 3 (07 2014), 66–73.

[43] Xinwen Zhang, Katharine Chang, Huijun Xiong, Yonggang Wen, Guangyu Shi, and Guoqiang Wang. 2011. Towards name-based trust and security for content-centric network. In *2011 19th IEEE International Conference on Network Protocols*. IEEE, 1–6.

[44] Philip R. Zimmermann. 1995. *The Official PGP User's Guide*. MIT Press, Cambridge, MA, USA.