ANDāNA: Anonymous Named Data Networking Application

Steven DiBenedetto
Colorado State University
dibenede@cs.colostate.edu

Paolo Gasti Gene Tsudik
University of California, Irvine
{pgasti,gtsudik}@uci.edu

Ersin Uzun
Palo Alto Research Center
ersin.uzun@parc.com

Abstract

Content-centric networking — also known as information-centric networking (ICN) — shifts emphasis from hosts and interfaces (as in today's Internet) to data. Named data becomes addressable and routable, while locations that currently store that data become irrelevant to applications.

Named Data Networking (NDN) is a large collaborative research effort that exemplifies the content-centric approach to networking. NDN has some innate privacy-friendly features, such as lack of source and destination addresses on packets. However, as discussed in this paper, NDN architecture prompts some privacy concerns mainly stemming from the semantic richness of names. We examine privacy-relevant characteristics of NDN and present an initial attempt to achieve communication privacy. Specifically, we design an NDN add-on tool, called ANDāNA*, that borrows a number of features from Tor. As we demonstrate via experiments, it provides comparable anonymity with lower relative overhead.*

1 Introduction

Although the Internet, as a whole, is a huge global success story, it is showing clear signs of age. In the 1970s, when core ideas underlying today's Internet were developed, telephony was the only example of effective global-scale communications. Thus, while the communication solution offered by the Internet's TCP/IP suite was unique and ground-breaking, the communication paradigm it focused on was similar to that of telephony: a point-to-point conversation between two entities. The communication world has changed dramatically since then and today's Internet has to accommodate: information-intensive services, exabytes of content created and consumed daily over the Web as well as a menagerie of mobile devices connected to it. To keep pace with these changes and move the Internet into the future, a number of research efforts to design new Internet architectures have taken off in the last few years.

Named-Data Networking (NDN) [32] is one such effort that exemplifies the content-centric approach [23, 27, 28] to networking. NDN names content instead of locations (i.e., hosts or interfaces) and thus transforms content into a first-class entity. NDN also stipulates that each piece of content must be signed by its producer. This allows decoupling of trust in content from trust in the entity that might store and/or disseminate that content. These NDN features facilitate automatic caching of content to optimize bandwidth use and enable effective simultaneous utilization of multiple network interfaces.

However, NDN introduces certain challenges that must be addressed in order for it to be a serious candidate for the future Internet architecture. One major argument for a new architecture is the inadequate level of security and privacy in today's Internet. We view anonymity as being a critical feature in any new network architecture. It helps people overcome communication restrictions and boundaries as well as evade censorship. In addition, some applications (e.g., e-cash or anonymous publishing) can be successfully deployed only if the underlying network allows users to hide their identity [14]. Even if end-users do not care about anonymity with respect to services they access, they might still want to hide their activities from employers, governments and ISPs, since those might censor, misuse or accidentally leak sensitive information [19].

Lack of source/destination addresses in NDN helps privacy, since NDN packets carry information only about *what* is requested but not *who* is requesting it. However, a closer look reveals that this is insufficient. In particular, NDN design introduces three important privacy challenges:

1. **Name privacy**: NDN content names are incentivized to be semantically related to the content itself. Similar to HTTP headers, names reveal significantly more information about content than IP addresses. Moreover, an observer can easily determine when two requests refer to the same (even encrypted) content.
2. **Content privacy**: NDN allows any entity that knows a name to retrieve corresponding content. Encryption in NDN is used to enforce access con-

trol and is not applied to publicly available content. Thus, consumers wanting to retrieve public content cannot rely on encryption to hide what they access.

3. **Cache privacy**: as with current web proxies, network neighbors may learn about each others' content access using timing information to identify cache hits.
4. **Signature privacy**: since digital signatures in NDN content packets are required to be publicly verifiable, identity of a content signer may leak sensitive information.

In this paper, we attempt to address these challenges. We present an initial approach, called ANDāNA that can be viewed as an adaptation of onion routing to NDN. Our approach is in-line with NDN principles. It is designed to take advantage of NDN strengths and work around its weaknesses. We optimized ANDāNA for small- to medium-size interactive communication – such as web-browsing and instant messaging – that are characterized by moderate amounts of low-latency traffic [11].

We provide a security analysis of the proposed approach under a realistic adversarial model. Specifically, we define anonymity and unlinkability under this security model and show that these properties hold for ANDāNA. Moreover, ANDāNA is secure with fewer anonymizing router hops than Tor. We prototyped ANDāNA and assessed its performance via experiments over a network testbed. Results show that ANDāNA introduces less overhead than Tor, especially, for anticipated traffic patterns.

We believe that this work is both timely and important. The former – because of the recent surge of interest in content-centric networking and NDN being a good example of this paradigm. (Also, while NDN is sufficiently mature to have a functional prototype suitable for experimental use, it is still at an early enough stage to be open to change.) The latter – because it represents the first attempt to identify and address privacy problems in a viable candidate for the future Internet architecture.

Before discussing details of our approach, we present further motivation for this work.

Why NDN? There are multiple efforts to develop new content-centric architectures and NDN is only one of those. We focus on NDN because it stands out in several aspects. First, it combines some revolutionary ideas about content-based routing that have attracted considerable attention from the networking research community. Second, it builds upon an open-source code-base called CCNx [12], that is led and continuously maintained by an industrial research lab (PARC). At the time of this writing (summer 2011), NDN is one of the very few content-centric architectural proposals with a reasonably mature prototype available to the research community.[1] Third, NDN is one of only four projects selected by NSF Future Internet Architectures (FIA) program [20].

On the other hand, NDN is an on-going research project and is thus subject to continuous change. However, we believe that it represents a good example of content-centric networking design and at least some of its concepts will influence the future of networking. More importantly, ideas, techniques and analysis discussed in this paper are not specific, or limited to, NDN; they are applicable to a wide range of designs, including host-, location- and content-addressable networks.

Approach. NDN follows the proven design principle of IP and claims to be the "thin waist" of the communications protocol stack. Thus, pushing security or privacy services (that are not critical for all types of communication) into this thin waist would contradict its design principle. Consequently, as in the case of IP, we believe that privacy tools should run on top of NDN. Looking at privacy and anonymity techniques in today's Internet, one well-established approach is an overlay anonymization network, exemplified by Tor [18]. Tor and its relatives employ layers of concentric encryption and intermediate nodes responsible for peeling off layers as packets travel through the overlay. This is commonly referred to as onion routing. Our approach falls into roughly the same category. However, as we discover and discuss in this paper, the task of adapting an anonymization overlay approach to NDN is not as simple as it might initially seem.

Scope. The primary focus of this paper is privacy. Security and other features of NDN are taken as given without justifying their existence. A number of important NDN-related security topics are out of scope of this paper, including: trust management, certification and revocation of credentials as well as routing security.

Organization. We start with NDN overview and privacy analysis in Section 2. Section 3 summarizes related work, followed by the description of ANDāNA in Section 4. Section 5 introduces a formal model for provable anonymity and security analysis of ANDāNA. Implementation details and performance evaluation results are discussed in Section 6. The paper concludes in Section 7.

2 NDN Overview

NDN [32] is a communication architecture based on named content.[2] Rather than addressing content by its location, NDN refers to it by name. Content name is composed of one or more variable-length components that are opaque to the network. Component

[1] We are aware of only two other content-centric architecture proposals – [33] and [36] – that have public prototypes.

[2] Note that we use the terms "content" and "data" interchangeably throughout this paper.

boundaries are explicitly delimited by "/". For example, the name of a CNN news content might be: /ndn/cnn/news/2011aug20. Large pieces of content can be split into fragments with predictable names: fragment 137 of a YouTube video could be named: /ndn/youtube/videos/video-749.avi/137.

Since the main abstraction is content, there is no explicit notion of "hosts" in NDN. (However, their existence is assumed.) Communication adheres to the *pull* model: content is delivered to consumers only upon explicit request. A consumer requests content by sending an *interest* packet. If an entity (a router or a host) can "satisfy" a given interest, it returns the corresponding *content* packet. Interest and content are the only types of packets in NDN. A content packet with name X in NDN is **never** forwarded or routed unless it is preceded by an interest for name X.[3]

When a router receives an interest for name X and there are no pending interests for the same name in its PIT (Pending Interests Table), it forwards this interest to the next hop according to its routing table. For each forwarded interest, a router stores some state information, including the name in the interest and the interface on which it was received. However, if an interest for X arrives while there is an entry for the same name in the PIT, the router collapses the present interest (and any subsequent ones for X) storing only the interface on which it was received. When content is returned, the router forwards it out on all interfaces where an interest for X has been received and flushes the corresponding PIT entry. Note that, since no additional information is needed to deliver content, an interest does not carry a source address. More detailed discussion of NDN routing can be found in [27].

In NDN, each network entity can provide content caching, which is limited only by resource availability. For popular content, this allows interests to be satisfied from cached copies distributed over the network, thus maximizing resource utilization. NDN deals with content authenticity and integrity by making digital signatures mandatory on all content packets. A signature binds content with its name, and provides origin authentication no matter how or from where it is retrieved. NDN calls entities that publish new content *producers*. Whereas, as follows from the above discussion, entities that request content are called *consumers*. (Consumers and producers are clearly overlapping sets.) Although content signature verification is optional in NDN, a signature must be verifiable by any NDN entity. To make this possible, content packets carry additional metadata, such as the ID of the content publisher and information on locating the public key needed for verification. Public keys are treated as regular content: since all content is signed, each public key content is effectively a "certificate". NDN does not mandate any particular certification infrastructure, relegating trust management to individual applications.

Private or restricted content in NDN is protected via encryption by the content publisher. Once content is distributed unencrypted, there is no mechanism to apply subsequent encryption. Specific applications may provide a means to explicitly request encryption of content by publishers. However, NDN does not currently allow consumers to selectively conceal content corresponding to their interests.

From the privacy perspective, lack of source and destination addresses in NDN packets is a clear advantage over IP. In practice, this means that the adversary that eavesdrops on a link close to a content producer can not immediately identify the consumer(s) who expressed interest in that content. Moreover, two features of standard NDN routers: (1) content caching and (2) collapsing of redundant interests, reduce the utility of eavesdropping near a content producer since not all interests for the same content reach its producer.

On the other hand, NDN provides no protection against an adversary that monitors local activity of a specific consumer. As most content names are expected to be semantically relevant to content itself, interests can leak a lot of information about the content they aim to retrieve. To mitigate this issue, NDN allows the use of "encrypted names", whereby a producer encrypts the tail-end (a few components) of a name [27]. [4] However, this simple approach does not provide much privacy: the adversary can link multiple interests for the same content – or those sharing the same name prefix – issued by different consumers. Moreover, an adversary can always replay an interest to see what (possibly cached) content it returns, even if a name of content is not semantically relevant.

3 Related Work

The goal of anonymizing tools and techniques is to decouple actions from entities that perform them. The most basic approach to anonymity is to use a trusted anonymizing proxy. A proxy is typically interposed between a sender and a receiver in order to hide identity of the former from the latter. The Anonymizer [3] and Lucent Personalized Web Assistant [22] are examples of this approach. While relatively efficient, it is susceptible to a (local) passive adversary that monitors all proxy ac-

[3] Strictly speaking, content named $X' \neq X$ can be delivered in response to an interest for X but only if X is a prefix of X'. As an example, the full name of each content packet contains the hash of that content; however, this hash value is usually not known to consumers and is typically omitted from interests.

[4] For example, a name such as: /ndn/xerox/parc/Alice/family/photos/Hawaii might be replaced with /ndn/xerox/parc/Alice/**encrypted-part**.

tivity. Also, a centralized proxy necessitates centralized (global) trust and represents a single point of failure.

A more sophisticated decentralized approach is used in mix networks [13]. Typically, a mix network achieves anonymity by repeatedly routing a message from one proxy to another, such that the message gradually loses any relationship with its originator. Messages must be made unintelligible to potentially untrusted intermediate nodes. Chaum's initial proposal [13] defines an anonymous email system, wherein a sender envelops a message with several concentric layers of public key encryption. The resulting message is then forwarded to a sequence of *mix* servers, that gradually remove one layer of encryption at a time and forward the message to the next mix server.

Subsequent research generally falls into two classes: delay-tolerant applications (e.g. email, file sharing) and real-time or low-latency applications (e.g. web browsing, VoIP, SSH). These two classes achieve different tradeoffs between performance (in terms of latency and bandwidth) and anonymity. For example, Babel [24], Mixmaster [30] and Mixminion [16] belong to the first category. Their goal is to provide anonymity with respect to the *global eavesdropper* adversary. Each mix introduces spurious traffic and randomized traffic delays in order to inhibit correlation between input and output traffic. However, unpredictable traffic characteristics and high delays make these techniques unsuitable for many applications.

Low-latency anonymizing networks are at the other end of the spectrum. They try to minimize extra latency by forwarding traffic as fast as possible. Because of this, strategies used in anonymization of delay-tolerant traffic – batching (delaying) and re-ordering of traffic in mixes, as well as introduction of decoy traffic — are generally not applicable. For example, [40] shows how traffic patterns can be used for de-anonymization in low-latency anonymity systems. Notable low-latency tools are summarized below.

Crowds [37] is a low-latency anonymizing network for HTTP traffic. It differs from traditional mix-based approaches as it lacks layered encryption. For each message it receives, an anonymizer probabilistically chooses to either forward it to a random next hop within the Crowds network or deliver it to its final destination. Since messages are not encrypted, Crowds is vulnerable to local eavesdroppers and predecessor attacks [43].

Morphmix [38, 39] is a fully distributed peer-to-peer mix network that uses layered encryption. Unlike Crowds, it does not require a lookup service to keep track of all participating nodes. Senders selects the first anonymizer and each anonymizer along an "anonymous tunnel" picks the next hop to dynamically build tunnels. Tarzan [21] is another fully distributed peer-to-peer mix network. It builds a universally verifiable set of neighbors (called mimics) for every node to keep track of other other Tarzan participants. Every node selects its mimics pseudo-randomly.

Tor [18] is the best-known and most-used low-latency anonymizing tool. It is based on onion routing and layered encryption. Tor uses a central directory to locate participating nodes and requires users to build a three-hop anonymizing circuit by choosing three random nodes. The first is called the *guard*, the second – the *middle*, and the third — *exit* node. Once set up, each circuit in Tor lasts about 10 minutes. For better performance, bandwidth available to nodes is taken into account during circuit establishment and multiple TCP connections are multiplexed over one circuit. Communication between Tor nodes is secured via SSL. However, Tor does not introduce any decoy traffic or randomization to hide traffic patterns. Another anonymization tool, I2P [26], adopts many ideas of Tor, while using a distributed untrusted directory service to keep track of its participants. I2P also replaces Tor's circuit-switching operation with packet-switching to achieve better load balancing and fault-tolerance.

A consumer privacy technique for Information-Centric Networks (ICNs) is proposed in [4]. Instead of using encryption, it leverages cooperation from content producers and requires them to mix sensitive information with so-called "cover" content. This approach requires producers to cooperate and store a large amount of cover traffic. It also does not provide consumer-producer unlinkability or protection against malicious producers.

Telex [44] is an alternative to mix networks designed to evade state-level censorship. It uses steganographic techniques to hide messages in SSL handshakes. Users connect to innocuous-looking unblocked websites through SSL. Sympathetic ISP-s that forward user's traffic recover hidden messages and deliver them to the intended destination. While novel, this approach presents significant deployment challenges and requires support from the network infrastructure. Furthermore, the threat model in Telex is quite different from that of the other anonymizing tools presented above. Moreover, established TCP fingerprinting techniques can easily detect differences between a Telex station and a censored website. Another analogous technique – called Cirripede [25] – was recently proposed.

4 ANDāNA

ANDāNA is a onion routing overlay network, built on top of NDN, that provides privacy and anonymity to consumers. In particular, ANDāNA prevents adversaries from linking consumers with the content they are retrieving. Following the terminology introduced in [37], ANDāNA provides *beyond suspicion*[5] degree of anonymity to its users.

ANDāNA uses multiple concentric layers of encryption and routes messages from consumers through a chain of at least two onion routers. Each router removes a layer of encryption and forwards the decrypted messages to the next hop. Due to its low-latency focus, ANDāNA does not guarantee privacy in presence of a global eavesdropper. However, since it is geared for a world-wide (or at least geographically distributed) network spanning a multitude of administrative domains, the existence of such an adversary is unlikely. For this reason, we restrict the adversarial capabilities to eavesdropping on, injecting, removing or modifying messages on a subset of available links. An adversary can compromise NDN routers and ANDāNA nodes at will. Nonetheless, consumers benefit from anonymity as long as they use at least one non-compromised ANDāNA node. Details of our adversarial model and formal privacy guarantees are discussed in Section 5.

4.1 Design

We now present two techniques — *asymmetric* and *session-based* — that provide privacy and anonymity for NDN traffic. Traffic is routed through *ephemeral circuits*, that are defined as a pair of distinct anonymizing routers (ARs). An AR is a NDN node (e.g. a router or a host) that chooses to be part of ANDāNA. An ephemeral circuit transports only one (or only a few) encrypted interest(s). It disappears either when the corresponding content gets delivered, or after a short timeout (hence "*ephemeral*"). A timeout interval is needed so that the consumer can re-issue the same encrypted interest in case of packet loss. We refer to the first AR as *entry router* and the second – as *exit router*. They must not belong to the same administrative domain and must not share the same name prefix. Optionally, consumers can select ARs according to some parameters, such as advertised bandwidth, availability or average load. As pointed out in [5, 31], there is a well know natural tension between non-uniform (i.e. performance-driven) choice of routers and anonymity. Consumers should consider this when selecting ARs.

To build an ephemeral circuit, a consumer retrieves the list of ARs and corresponding public keys. Although we do not mandate any particular technique, a consumer can retrieve this list using, e.g., a directory service [18] or a decentralized (peer-to-peer) mechanism. AR public keys can be authenticated using decentralized techniques (such as web-of-trust [2]) or a PKI infrastructure.[6]

A prospective AR joins ANDāNA by advertising its public key, together with its identity defined as: namespace, organization and public key fingerprint. An AR also publishes auxiliary information, such as total bandwidth, average load, and uptime.

As mentioned earlier, both interest and content packets leak information. Even if names in interests are hidden, three components of content packets — signatures, names and content itself — contain potentially sensitive information. Of course, content producers could simply generate a new key-pair to sign each content packet. This would be impractical, since high costs of key generation and distribution would make it difficult for consumers to authenticate content. (Note that key-evolving schemes [8] do not help, since verification keys generally evolve in a way that is predictable to all parties, including the adversary.) Alternatively, the original content signature could be replaced with that generated by an AR. However, this would preclude end-to-end content verifiability and thus break the NDN trust model.

For this reason, ANDāNA implements encrypted encapsulation of original content, using two symmetric keys securely distributed by the consumer to the ARs during setup of the ephemeral circuit. Upon receiving a content packet, the exit router encrypts it, together with the original (cleartext) name and signature, under the first key provided by the consumer. Then, treating the ciphertext as payload for a new content packet, the exit router signs and sends it to the entry router. The latter strips this signature and the name and encrypts the remaining ciphertext under the second symmetric key provided by the consumer. Next, it forwards the ciphertext with the original encrypted name and a fresh (its own) signature. After decrypting the payload, the consumer discards the signature from the entry router and verifies the one from the content producer.

Because decryption is deterministic, an encrypted interest sent to an AR always produces the same output. Since ARs are a public resource, the adversary can use them to decrypt previously observed interests. It can thus observe the corresponding output and correlate incoming/outgoing interests. This is a well-known attack and there are several ways to mitigate it, such as encrypted channels between communicating parties [18] and mixing (for delay-tolerant traffic) [24]. However, such techniques tend to have significant impact on computational costs and latency. Instead, we use standard NDN features of interest aggregation and caching to prevent such attacks, as described next.

In NDN, a router (not just an AR) that receives duplicate interests collapses them. An interest is considered a *duplicate*, if it arrives while another interest referring

[5] For any packet observed by the adversary, an entity is considered *beyond suspicion* if it is as likely to be the sender of this packet as any other entity.

[6] Note that implicit replication implemented through caching allows the construction of a directory system with better resilience against denial-of-service (DoS) attacks than IP.

to the same content has not been satisfied. Also, if the original interest has been satisfied and the corresponding content is still in cache, a new interest requesting the same piece of data is satisfied with cached content. In this case, the router does not forward any interests. Therefore, the adversary must wait for the expiration of cached content.

As part of ANDāNA, the consumer includes its current timestamp within each encryption layer. ARs reject interests with timestamps outside a pre-defined time window. Thus, consumers need to be loosely synchronized with ARs that must reserve at least $(rate \times window)$ of cache, where $rate$ is the router's wire-rate and $window$ is the interval within which interests are accepted. In this way, if an interest is received multiple times by an AR (e.g. in case of loss of the corresponding data packet between the AR and the consumer), the AR is able to satisfy it using its cache.

The encryption algorithm used by consumers to conceal names in interests must be secure against adaptive chosen ciphertext (CCA) attacks.[7] CCA-security [9] implies, among other things, probabilistic encryption and non-malleability. The former prevents the adversary from determining whether two encrypted interests correspond to the same unencrypted interest. Whereas, the latter implies that the adversary cannot modify interests to defeat the mechanism described above.

We now describe two flavors of anonymization protocols: asymmetric and session-based. In order to allow efficient routing of interest packets, the encrypted component is encoded at the end of the name with both flavors.

Asymmetric: To issue an interest, a consumer selects a pair of ARs and uses their public keys to encrypt the interest, as described above and in Algorithm 1. A consumer also generates two symmetric keys: k_1 and k_2 that will be used to encrypt the content packet on the way back. We use $\mathcal{E}_{pk}(\cdot)$ and $\overline{\mathcal{E}_k}(\cdot)$ to denote (CCA-secure) public key and symmetric encryption schemes, respectively.

To account for the delay due to extra hops needed to reach the second AR (and reduce the number of discarded interests), a consumer adds half of the estimated round trip time (RTT) to the innermost timestamp. Each AR removes the outermost encryption layer, as detailed in Algorithm 2. Since $\mathcal{E}_{pk}(\cdot)$ is CCA-secure, the decryption process fails if the ciphertext has been modified in transit or was not encrypted under the AR's public key. Content corresponding to the encrypted interest is encrypted on the way back, as detailed in Algorithm 3, using $\overline{\mathcal{E}_k}(\cdot)$ and symmetric keys supplied by the consumer.

Algorithm 1: Encrypted Interest Generation

```
input : Interest int;  Set of $\ell$ ARs and their keys:
        $\mathcal{R} = \{(\mathrm{AR}_i, pk_i) \mid 0 < i \leq \ell, pk_i \in \mathcal{PK}\}$
output: Encrypted interest $\mathsf{int}_{pk_i,pk_j}$;  symmetric keys $k_1, k_2$
1: Select $(\mathrm{AR}_i, pk_i), (\mathrm{AR}_j, pk_j)$ from $\mathcal{R}$
2: if $\mathrm{AR}_i = \mathrm{AR}_j$ or $\mathrm{AR}_i$, $\mathrm{AR}_j$ are from same organization or
   $\mathrm{AR}_i$, $\mathrm{AR}_j$ share the same name prefix then
3:     Go to line 1
4: end if
5: $k_1 \leftarrow \{0,1\}^\kappa$ ;  $k_2 \leftarrow \{0,1\}^\kappa$
6: $eint = \mathrm{AR}_2/\mathcal{E}_{pk_j}(\mathsf{int} \mid k_2 \mid curr\_timestamp + RTT/2)$
7: $eint = \mathrm{AR}_1/\mathcal{E}_{pk_i}(eint \mid k_1 \mid curr\_timestamp)$
8: Output $eint, k_1, k_2$
```

Algorithm 2: AR Handling of Encrypted Interests

```
input : Encrypted Interest $\mathsf{int}_{pk_i,pk_j}$, where
        $pk_i, pk_j \in \mathcal{PK} \cup \{\perp\}$  (where "$\perp$" denotes "no
        encryption")
output: Interest $\mathsf{int}_{pk_j}$;  symmetric key $k_1$
1: $(\mathsf{int}_{pk_j}, k_1, timestamp) = \mathcal{D}_{sk_i}(\mathsf{int}_{pk_i,pk_j})$
2: if Step 1 fails or $timestamp$ is not current then
3:     Discard $\mathsf{int}_{pk_i,pk_j}$
4: else
5:     Save tuple $(\mathsf{int}_{pk_i,pk_j}, \mathsf{int}_{pk_j}, k_1)$ to internal state
6:     Output $\mathsf{int}_{pk_j}, k_1$
7: end if
```

Algorithm 3: AR Content Routing

```
input : Content: $data_{k_2}$ in response to $\mathsf{int}_{pk_j}$, where
        $pk_j \in \mathcal{PK} \cup \{\perp\}$
output: Encrypted data packet $data_{k_1,k_2}$
1: Retrieve tuple $(\mathsf{int}_{pk_i,pk_j}, \mathsf{int}_{pk_j}, k_1)$ from internal state
   where name in $\mathsf{int}_{pk_2}$ matches that in $data_{k_2}$
2: if $k_2 \neq \perp$ then Remove signature and name from $data_{k_2}$
3: Create new empty data packet $pkt$
4: Set name on $pkt$ as the name on $\mathsf{int}_{pk_i,pk_j}$
5: Set the data in $pkt$ as $\overline{\mathcal{E}_{k_1}}(data_{k_2})$
6: Sign $pkt$ with AR's key
7: Output $pkt$ as $data_{k_1,k_2}$
```

Session-based Variant. This variant aims to reduce (amortize) the use of public key encryption thus lowering the computational cost and ciphertext size. Before sending any interests through ephemeral circuits, a consumer (Alice) establishes a shared secret key with each selected AR. This is done via a 2-packet interest/content handshake. We do not describe the details of symmetric key setup, since there are standard ways of doing it. We provide two options: one using Diffie-Hellman key exchange [17], and the other – using SSL/TLS-style protocol whereby Alice encrypts a key for AR_i. Once a symmetric key k_{ai} is shared with AR_i, Alice can establish any number of ephemeral circuits using it as either first or second AR hop. Also at setup time, Alice and AR_i agree on session identifier value – sid_{ai} – that is included (in cleartext) in subsequent interests so that AR_i

[7] Technically, in order to guarantee correctness an encryption scheme suitable for ANDāNA must also be robust [1]. However, since CCA-secure encryption schemes used in practice are also robust, we omit this requirement in the rest of the paper.

can identify the appropriate entry for Alice and k_{ai}.

The main advantage of the session-based approach is better performance: both consumers and routers only perform symmetric operations after initial key setup. However, one drawback is that, since the session identifier sid is not encrypted, packets corresponding to the same sid are easily linkable.

We note that our design neither encourages nor prevents consumers from mixing asymmetric *and* session-based variants for the same or different ephemeral circuits.

4.2 System and Security Model

In order for our discussion to relate to prior work, we use the notion of "indistinguishable configurations" from the framework introduced in [19]; the actual definitions are in Section 5.

Our security analysis considers the worst case scenario, i.e., interests being satisfied by the content producer rather than a router's cache. While, in normal conditions, encrypted interests are satisfied by caches only in case of packet loss, fully decrypted interests may not have to reach to content producers. A system secure in case of cache misses is also secure when interests are satisfied by content cached at routers along the way. (Recall that, when an interest is satisfied by a router's cache, it is not forwarded any further.) This limits the adversary's ability to observe interests in transit.

Adversary Goals and Capabilities. The goal of an adversary is to link consumers with their actions. In particular, it may want to determine what content is being requested by a particular user and/or which users are requesting specific content. A somewhat related goal is determining which cache (if any) is satisfying a consumer's requests. Our adversary is local and active: it controls only a subset of network entities and can perform any action usually allowed to such entities. Moreover, it is capable of selectively compromising additional network entities according to its local information.

Our model allows the adversary to perform the following actions:

- **Deploy compromised routers**: ANDāNA is an open network, therefore an adversary can deploy compromised anonymizers and regular routers. As such, routers may exhibit malicious behavior including injection, delay, alteration, or drop traffic.
- **Compromise existing routers**: An adversary can select any router (either ARs or regular routers) in the network and compromise it. As a result, the adversary learns all the private information (e.g. decryption keys, pending decrypted interests, cache content, etc.) of such router.
- **Control content producers**: Content producers are not part of ANDāNA. As such, the network has no control over them. An adversary can compromise existing content producers or deploy compromised ones and convince users to pull content from them. We also assume that the content providers are publicly accessible, and therefore the adversary is able to retrieve content from them.
- **Deploy compromised caches**: Similarly to compromised content producers, an adversary can compromise routers' cache or deploy its own caches. The behavior of a compromised cache includes monitoring cache requests and replying with corrupted data.
- **Observe and replay traffic**: An adversary can tap a link carrying anonymized traffic. By doing this it learns, among other things, packet contents and traffic patterns. The traffic observed by an adversary can be replayed by any compromised router.

An adversary can iteratively compromise entities of its choice, and use the information it gathers to determine what should be compromised next. In order to make our model realistic, the time required by an adversary to compromise or deploy a router, a cache or a content producer is significantly higher that the round-trip time (RTT) of an anonymized interest and corresponding data. This implies that all the state information recovered from a newly compromised router only refers to packets received *after* the adversary decides to compromise such router.

A powerful class of attacks against anonymizing networks is called fingerprinting [29, 41]. Inter-packet time intervals are usually not hidden in low latency onion routing networks because packets are dispatched as quickly as possible. This behavior can be exploited by an adversary, who can correlate inter-packet intervals on two links and use this information to determine if the observed packets belong to the same consumer [41]. This class of attacks is significantly harder to execute on ANDāNA because of the nature of ephemeral circuits and because of the use of caches on routers. Ephemeral circuits do not allow the adversary to gather enough packets with uniform delays since they are used to transport only one or a very small number of interests and corresponding data. Active adversaries who can control the communication link of a content provider can add measurable delays to some of the packets in order to identify consumers. However, consumers may be able to retrieve the same content through caches making such attack ineffective. Throughput fingerprinting consists in measuring the throughput of the circuit used by a consumer to identify the slowest anonymizer in the consumer's circuit [29]. Throughput fingerprinting is difficult to perform in ANDāNA since each ephemeral circuit does not carry enough information to mount an attack. In particular, the authors of [29] report that a successful attack requires at least a few minutes of traffic on Tor. Similarly, ephemeral circuits provide an effective protection against known attacks such as the predecessor attack [43].

Consumers, Producers and ARs. Each consumer runs several processes that generate interests. For our analysis, interests are created by a specific interface of a host, and the corresponding content is delivered back to the same interface. Interest encryption is either performed on the consumer's host, or on an entity that routes consumer's traffic. In the latter case, the channel between the user and the anonymizing entity is considered secure.

Content is generated by producers, i.e., entities that can sign data. We do not assume the correspondence between a producer and a particular host. Content can be either stored in routers' caches, at servers or dynamically generated in response to an interest.

ARs perform interests decryption and content encapsulation. Each AR advertises a public key for signature verification and one or more public keys for encryption. ARs must refresh their encryption keys frequently, discarding old keys after a short grace period. In order to simplify key distribution and allow consumer to immediately trust new public keys from routers, we use a simple key hierarchy where a long lived public key owned by the router (the signing key), is used to certify short lived encryption keys. The signing key may be certified by other entities using techniques like web-of-trust or PKI.

Denial-of-service Attacks. ANDāNA is envisioned as a public overlay network and is clearly susceptible to DoS attacks. Since anyone can join ANDāNA as an AR or use it as a consumer, we make no distinction between insider and outsider attacks. The adversary can send numerous interests to ARs or construct ephemeral circuits longer than two hops in order to maximize effectiveness of attacks. Moreover, it can consume AR resources by sending malformed encrypted interests that require ARs to perform expensive and ultimately useless public key decryption. Similar to Tor, before establishing an ephemeral circuit, an AR can ask a consumer to solve an easy-to-verify/expensive-to-solve puzzle. This and similar techniques for ANDāNA are subjects of future work. In a setting with long-lived circuits, such as Tor, disrupting a node effectively shuts down all circuits that include it. Due to the short lifespan of our ephemeral circuits, the same attack on ANDāNA only causes a very small number of interests/data packets per user to be dropped.

Abuse. Similar to any other anonymity service, ANDāNA can be abused for a variety of nefarious purposes. We do not elaborate on this topic. However, exit policies similar to those in Tor [18] can be used with ANDāNA based on content names.

5 Security Analysis

In this section we propose a formal model for evaluating the security of ANDāNA. We define consumer anonymity and unlinkability with respect to an adversary within this model. We finally provide necessary and sufficient conditions for anonymity and unlinkability. As our analysis shows, we are able to obtain a level of anonymity comparable to Tor with two — rather than Tor's three — ARs thanks to the lack of source addresses in NDN interests.

In general, efficacy of ANDāNA depends on the inability of the adversary to correlate input and output of a non-compromised AR, and its inability to observe all producer and consumers at the same time. Since ANDāNA is designed for low-latency traffic, we do not intentionally delay messages or introduce dummy packets, other than some limited padding. This is similar to how Tor and other low-latency anonymizing networks forward traffic, and implies that traffic patterns remain almost unchanged as they pass through the network [31]. It is well known that, in Tor, this allows the adversary that observes both ends of a communication flow to confirm a suspected link between them [5, 35]. For this reason, a *global passive adversary* can violate anonymity properties of both Tor and ANDāNA. However, we believe that such an adversary is unrealistic in a geographically distributed network spanning over multiple administrative domains, and designing against it would result in overkill.

We assume that any adversary monitoring all interfaces of an AR can correlate entering encrypted traffic with its exiting, decrypted counterpart using timing information. However, we believe that the short lifespan of ephemeral circuits – and therefore the limited number of related packets traveling through a single AR – severely limits the adversary's ability to carry out this attack. Unfortunately, at the time of this writing we do not have enough experimental evidence to confirm this. For the sake of safety, in the analysis below we assume that, by compromising all interfaces of an AR, the adversary also compromises the AR itself. Therefore, a non-compromised AR must have at least one non-compromised interface. To sum up, we assume that:

Assumption 5.1. *Adv cannot correlate input and output of a non-compromised AR.*

Our analysis is based on *indistinguishable configurations*. A configuration defines consumers' activity with respect to a particular network. *Adv* only controls a subset of network entities and observes only some packets. Therefore, it cannot distinguish between two configurations that vary only in the activity that it cannot directly observe or in the content of encrypted packets that it cannot decrypt. In order to provide meaningful anonymity guarantees, we identify a set of configurations that have one or more equivalent counterparts. However, unlike [19], our analysis takes into account the infrastructure underlying ANDāNA, i.e., the network topology and packets exchanged over the *actual* network. We believe that this makes our model and analysis both realistic and fine-grained, since it accounts for all adversarial advantages related to the underlying network structure. Packets sent by a non-compromised consumer u to a non-compromised AR r transit through several — possibly compromised — NDN routers that are not part of ANDāNA. The model of [19] considers r compromised even if only one link between u and r is controlled by the adversary. Whereas, in our model, r is considered to be non-compromised.

Notation and Definitions

Table 1 summarizes our notation. The intersection of P and C might not be empty, which reflects the fact that consumers can act as producers and *vice versa.* Similarly, our model does not prevent routers from being producers and/or consumers. Therefore, $\mathsf{R} \cap \mathsf{P}$ and $\mathsf{R} \cap \mathsf{C}$ might be non-empty.

The adversary is defined as a 4-tuple: $Adv = (\mathsf{P}_{Adv}, \mathsf{C}_{Adv}, \mathsf{R}_{Adv}, \mathsf{IF}_{Adv}) \subset (\mathsf{P}, \mathsf{C}, \mathsf{R}, \mathsf{IF})$ where individual components specify (respectively) sets of: compromised producers, consumers, routers and interfaces. If $r \in \mathsf{R}_{Adv}$, then Adv controls all interfaces and has access to all decryption key and state information of r. If all interfaces of r are in IF_{Adv}, then $r \in \mathsf{R}_{Adv}$. In other words, for the sake of this analysis, controlling all interfaces of a router is equivalent to learning that router's decryption/secret key. We emphasize that for $r \in \mathsf{R}$ to be non-compromised, at least one of its interfaces must be non-compromised. If $p \in \mathsf{P}_{Adv}$, Adv controls p's interfaces, monitors interests received by p and controls both content and timing of p's responses to incoming interests. If $c \in \mathsf{C}_{Adv}$, then Adv controls all fields and timing of interests. Finally, if $\mathsf{if} \in \mathsf{IF}_{Adv}$, then Adv can listen to all traffic flowing through if, as well as sending new traffic from it. IF_{Adv} includes all the interfaces of compromised consumers, producers and routers *plus* additional interfaces eavesdropped on by Adv.

For ease of notation, we do not explicitly indicate the name of the next router in interest packets nor symmetric keys chosen by consumers. We denote encrypted interests as:

$$\mathsf{int}_{pk_1,pk_2} = \mathcal{E}_{pk_1}(\mathcal{E}_{pk_2}(\mathsf{int}))$$

with $pk_1, pk_2 \in \mathcal{PK} \cup \{\perp\}$ where $\perp$ indicates a special symbol for "no encryption". If $pk_1 = \perp$ then $pk_2 = \perp$. The size of public keys is a function of the global security parameter κ. For simplicity, we denote $\mathsf{int}_{pk_1,\perp}$ as int_{pk_1}. When an AR receives int_{pk_1,pk_2} and if it is in possession of the decryption key corresponding to pk_1, it removes the outer layer of encryption. While $\mathcal{E}$ is CCA-secure (and therefore also CPA-secure), we do not require $\mathcal{E}$ to be key private [6]. Key privacy prevents an observer from learning the public key used to generate a ciphertext. In ANDāNA, knowledge of the public key used to encrypt the outer layer of an interest does not reveal any more information than the (cleartext) name on the interest.

We define the anonymity set with respect to interface if_i^r as:

$$A_{\mathsf{if}_i^r} = \{d \mid \Pr[d \rightarrow_{\mathsf{int}} r \mid \mathsf{int} \rightsquigarrow \mathsf{if}_i^r] > 0\}$$

In other words, for each interface if_i^r of router r, $A_{\mathsf{if}_i^r}$ contains all entities that could have sent int with non-zero probability. We define $\mathsf{path}^{\mathsf{int}} = \{\mathsf{if}_i^r \mid \mathsf{int} \rightsquigarrow \mathsf{if}_i^r\}$. This is the sequence of interfaces traversed by int. We use it to define the anonymity set of an interest with respect Adv:

$$A_{Adv}^{\mathsf{int}} \triangleq \bigcap_{\mathsf{path}^{\mathsf{int}} \cap \mathsf{IF}_{Adv}} A_{\mathsf{if}_i^r}$$

Intuitively, if u is far away from a compromised entity d, then all sets A_{Adv}^{int} such that $u \in A_{Adv}^{\mathsf{int}}$ are a large subset of C. Adv can rule out possible senders of an interest (i.e., determine if $u \notin A_{Adv}^{\mathsf{int}}$) only if it controls at least one entity (routers, interfaces) along each path that u does not share with other consumers. The level of anonymity of $u \in A_{Adv}^{\mathsf{int}}$ with respect to Adv is proportional to the size of A_{Adv}^{int}. In particular, if u is the only member of A_{Adv}^{int}, it has no anonymity, since int must have been issued by u.

A configuration is a description of the network activity. Each configuration maps consumers to their actions, defined as the interest they issue and the corresponding content producers. More formally, a configuration is a relation:

$$C : \mathsf{C} \rightarrow \{(r_1, r_2, p, \mathsf{int}_{pk_1,pk_2})\}$$

with $(r_1, r_2, p, \mathsf{int}_{pk_1,pk_2}) \in \mathsf{R}^2 \times \mathsf{P} \times \{0,1\}^*$, that maps a consumer to: a pair of routers defining an ephemeral circuit, an interest (encrypted for this circuit) and a producer. $C(u)$ is a 4-tuple that represents one action of u in C. C_i is the selection on the i-th component of C, i.e., if $C(u) = (r_1, r_2, p, \mathsf{int}_{pk_1,pk_2})$, then $C_1(u) = r_1$, $C_2(u) = r_2$, $C_3(u) = p$ and $C_4(u) = \mathsf{int}_{pk_1,pk_2}$.

We say that two configurations C and C' are "indistinguishable with respect to Adv" if Adv can only determine with probability at most $1/2 + \varepsilon$ which configuration corresponds to the observed network, for some ε negligible in the security parameter κ. We denote two such configurations as $C \equiv_{Adv} C'$.

We now show that assumption 5.1 holds if a passive adversary observes only input and output values of

C	set of all consumers, $u \in \mathsf{C}$	Adv	adversary
P	set of all content producers, $p \in \mathsf{P}$	d	an entity, i.e., a router or a host
R	set of all routers, $r \in \mathsf{R}$	$d \rightarrow_{\mathsf{int}} r$	entity d sends interest int to some interface of router r
IF	set of all interfaces on all network devices	$\mathsf{int} \rightsquigarrow \mathsf{if}_i^r$	router r receives interest int on interface if_i^r
$\mathsf{if}_i^r \in \mathsf{IF}$	i-th interface on router r	$\mathcal{E}_{pk}(\cdot)$	CCA-secure hybrid encryption scheme
$\mathcal{PK}$	set of all public keys	int_{pk_1,pk_2}	interest encrypted under public keys pk_1, pk_2
(pk_i, sk_i)	public/priv. encryption keypair of an AR r_i	$\perp$	no encryption

Table 1. Notation.

an AR (i.e., it cannot use timing information or other side-channels), and the underlying encryption scheme is semantically secure. Claim 5.1 below states that, for any encrypted interest, Adv cannot determine if it corresponds to an interest decrypted by a non-compromised router, by observing the two and with no additional information.

Claim 5.1. *Given any CPA-secure public key encryption scheme $\mathcal{E}$ and two same-length interests $\mathsf{int}^0, \mathsf{int}^1$ chosen by Adv, Adv has only negligible advantage over $1/2$ in determining the value of a randomly selected bit b, given $\mathsf{int}^b_{pk_1,pk_2}$, $\mathsf{int}^0_{pk_2}$ and $\mathsf{int}^1_{pk_2}$, with $pk_1 \in \mathcal{PK}$ and $pk_2 \in \mathcal{PK} \cup \{\perp\}$.*

Due to the lack of space, Claim 5.1 is formally justified in Appendix A.

Anonymity Definitions and Conditions

In this section we present formal definitions of anonymity for our model. We introduce the notions of *consumer anonymity*, *producer anonymity* and *producer and consumer unlinkability*. We show that ephemeral circuits composed of two anonymizing routers — at least one of which is not compromised — provide consumer *and* producer anonymity. This, in turn, implies consumer and producer unlinkability. Due to the lack of space, we defer formal proofs of the theorems in this section to Appendix A.

A consumer u enjoys *consumer anonymity* if Adv cannot determine whether u or a different user u' is retrieving some specific content. This notion is formalized using indistinguishable configurations: given a configuration C in which u retrieves content t, u has consumer anonymity if there exist another configuration C' in which u' retrieves t and Adv cannot determine whether he is observing C or C'. More formally:

Definition 5.1 (Consumer anonymity)**.** *$u \in (\mathsf{C} \setminus \mathsf{C}_{Adv})$ has consumer anonymity in configuration C with respect to Adv if there exists $C' \equiv_{Adv} C$ such that $C'(u') = C(u)$ and $u' \neq u$.*

Theorem 5.1. *$u \in (\mathsf{C} \setminus \mathsf{C}_{Adv})$ has consumer anonymity in C with respect to Adv if there exists $u' \neq u$ such that any of the following conditions hold:*

1. *$u, u' \in A^{C_4(u)}_{Adv}$*
2. *$C_1(u) = C_1(u')$, $C_1(u) \notin \mathsf{R}$ and $C_1(u) \in A^{\mathsf{int}_{pk_2}}_{Adv}$ where $C_4(u) = \mathsf{int}_{pk_1,pk_2}$*
3. *$C_2(u) = C_2(u')$, $C_2(u) \notin \mathsf{R}$ and $C_2(u) \in A^{\mathsf{int}}_{Adv}$ where $C_4(u) = \mathsf{int}_{pk_1,pk_2}$*

Informally, the theorem above states that ANDāNA provides consumer anonymity with respect to Adv if: *1.* Adv cannot observe encrypted interests coming from u and u', or it cannot distinguish between the two consumers due to anonymity provided by the network layer; or *2.* u, u' share an non-compromised first router in at least one ephemeral circuit; or *3.* u, u' share an non-compromised second router in at least one ephemeral circuit.

Similarly to consumer anonymity, producer anonymity is defined in terms of indistinguishable configurations. In particular, a producer p enjoys anonymity with respect to Adv which observes int_{pk_1,pk_2} if Adv cannot distinguish between a configuration C where p produces the content corresponding to int and a configuration C' where p' and not p produces *that* content.

Definition 5.2 (Producer anonymity)**.** *Given int_{pk_1,pk_2} for $p \in \mathsf{P}$, $u \in \mathsf{C}$ has producer anonymity in configuration C with respect to p, Adv if there exists an indistinguishable configuration C' such that int_{pk_1,pk_2} is sent by a non-compromised consumer to a producer different from p.*

Theorem 5.2. *u has producer anonymity in C with respect to p, Adv if any of the following conditions hold:*

1. *There exists $C(u)$ such that $C_1(u)$ (the first anonymizing router) is not compromised and $C_4(u) = \mathsf{int}_{pk_1,pk_2}$, $C_1(u) = C_1(u')$ and $C_3(u) = p \neq C_3(u')$ for some non-compromised $u' \in \mathsf{C}$,* or
2. *There exists $C(u)$ such that $C_2(u)$ (the second anonymizing router) is not compromised and $C_4(u) = \mathsf{int}_{pk_1,pk_2}$, $C_2(u) = C_2(u')$ and $C_3(u) = p \neq C_3(u')$ for some non-compromised $u' \in \mathsf{C}$*

Finally, we define producer and consumer unlinkability as:

Definition 5.3 (Producer and consumer unlinkability)**.** *We say that $u \in (\mathsf{C} \setminus \mathsf{C}_{Adv})$ and $p \in \mathsf{P}$ are unlinkable in C with respect to Adv if there exists $C' \equiv_{Adv} C$ where u's interests are sent to a producer $p' \neq p$.*

Corollary 5.1. *Consumer* $u \in (\mathsf{C} \setminus \mathsf{C}_{Adv})$ *and producer* $p \in \mathsf{P}$ *are unlinkable in configuration* C *with respect to* Adv *if* p *has producer anonymity with respect to* u*'s interests* or u *has consumer anonymity and there exists a configuration* $C' \equiv_{Adv} C$ *where* $C'(u') = C(u)$ *with* $u' \neq u$ *and* u'*'s interests have a destination different from* p*.*

Corollary 5.2. *Consumer* $u \in (\mathsf{C} \setminus \mathsf{C}_{Adv})$ *and producer* $p \in \mathsf{P}$ *are unlinkable in configuration* C *with respect to* Adv *if both producer and consumer anonymity hold.*

We emphasize that this result also holds for ephemeral circuits with length greater than two ARs.

6 Implementation and Performance

ANDāNA is implemented as an application-level service consisting of client "stack" (used by consumers) and server program that runs on ANDāNA ARs. Both are written in C and interface to NDN through Unix domain sockets.[8] Cryptographic algorithms are implemented using OpenSSL [42]. Hybrid encryption is obtained using RSA-OAEP [10] and AES+HMAC [15, 7]. The latter is also used for symmetric encryption. We use SHA-256 for HMAC and 1024- and 128-bit keys for RSA and AES, respectively. Loose time synchronization among ANDāNA client and servers are achieved using `pool.ntp.org`, a public pool of NTP servers.

ANDāNA client encrypts interests from user applications. In order to hide all possible sources of de-anonymizing information, encryption is performed over the full interest packet, including: name, scope, exclusion filters and duplicate suppression string fields. Following NDN "rules", ANDāNA AR announces the ability to serve the root ("`/`") namespace and receives all traffic sent from (or to) the local NDN routing process. This allows traffic to be routed through ANDāNA by default, requiring no changes to existing applications. For more granularity, consumers can vary the default namespace, e.g., "`/andana/`". However, this would require privacy-seeking applications to explicitly direct their traffic to that namespace, similar to today's configurable proxy settings.

ANDāNA servers run as applications on NDN routers. Each server is responsible for its *relay* and *session creation* namespaces. The former is a globally routable namespace used for receiving both session-based and asymmetrically encrypted Interests. Clients using session-based encryption in ANDāNA need to first establish symmetric keys with servers. To start a new session with a server, a clients sends an interest in the `createsession` namespace, registered by the server code as a sub-prefix of the relay namespace.

We deployed our prototype and run a series of tests on the Open Network Laboratory (ONL) [34]. ONL is a testbed developed by Washington University to enable experimental evaluation of advanced networking concepts in a realistic environment. To guarantee highly reproducible results, ONL provides reservation-based exclusive access to most of its host and network resources. All our experiments used single-core Linux machines with 512 MB of RAM and gigabit switches (one machine per switch).

We compare plain NDN and ANDāNA on a simple line topology with four switches and four Linux machines, each corresponding to an NDN node. Static routing is established between nodes. The first NDN node in the line topology acts as a consumer and runs `ccngetfile` — a small tool from CCNx open-source library that retrieves data published as NDN content and stores it in a local file. We performed tests with 1, 10, and 100MB files; each file was retrieved from the NDN repository of the machine at the other end of the line topology. Results of this comparison for 10MB files are summarized in Fig. 1. Due to space constraints, we illustrate all file retrieval results in Appendix B. Results show that computational overhead introduced by ANDāNA roughly doubles download times over plain NDN. This is assuming an almost-perfect world where ARs topologically align with the best path and link bandwidths are abundant.

In order to compare ANDāNA's computational overhead with a similar anonymizing tool, we deployed Tor over ONL and measured its overhead over TCP/IP. We measured performance of TCP/IP baseline deploying five switches, connected in a line, and two Linux machines (one at each end): the first acting as client (running `curl`), the second – as server (running `lighttpd` HTTP server). Performance of Tor was measured on a topology that closely mimics that of TCP/IP baseline: five switches, connecting three Tor relays, a client and a server. To ensure "line" topology, Tor client is configured to use explicit entry and exit nodes; DNS lookups are avoided by using IP addresses in all tests.

Before discussing the results, we mention some comparison details. NDN is a research project and its code is optimized for functionality rather than performance. It provides content authentication through digital signatures – a computationally expensive feature not present in either TCP/IP or Tor. NDN stack currently runs as a user-space application, in contrast to TCP/IP that runs in kernel-space. Finally, in all our experiments, NDN had to run on top of TCP/IP (rather than at layer 2) due to limitations of the underlying ONL testbed. Consequently, we believe a fair comparison between ANDāNA and Tor can only be achieved by focusing the analysis on *relative* overhead imposed by each, over the network

[8] At the time of this writing, there is no direct function interface to NDN

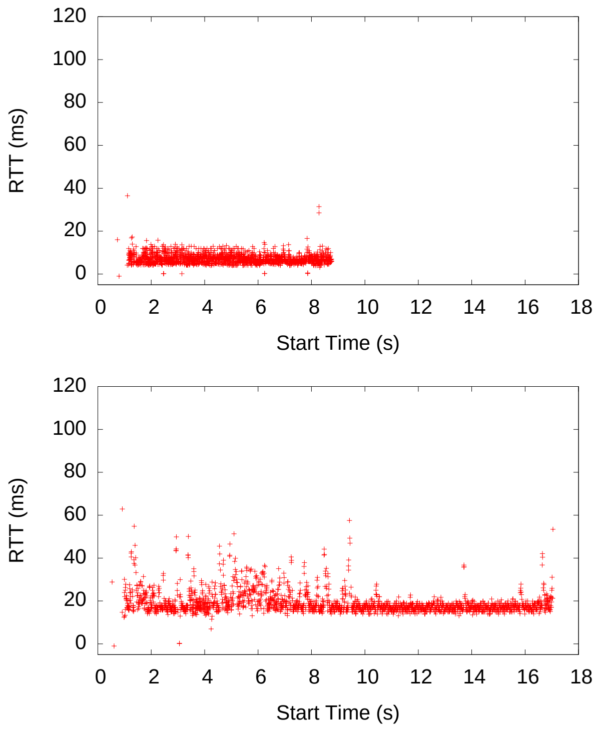

Figure 1. Left: RTT for 10MB of content over NDN (limited anonymity). Right: RTT for 10MB of content over ANDāNA (full anonymity).

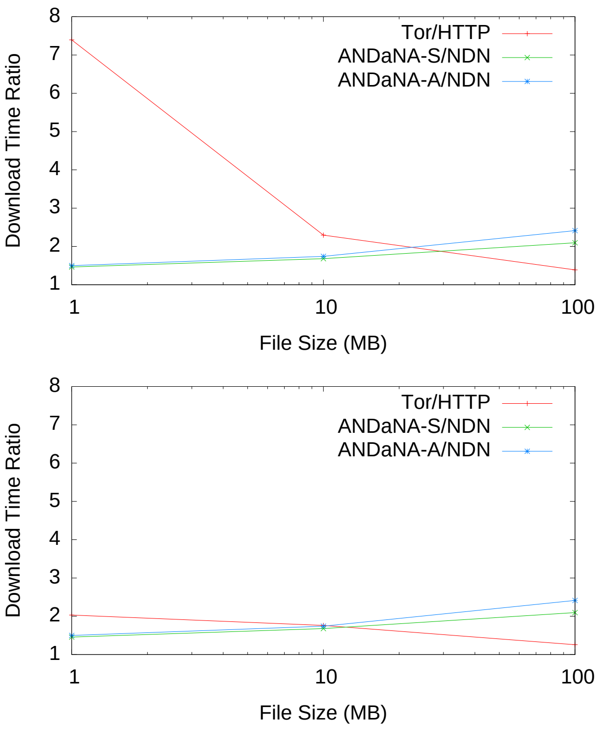

Figure 2. Comparison of 1, 10, and 100MB file download times over Tor, ANDāNA-S and ANDāNA-A with respect to respective baselines. Left: transfer time *and* circuit setup time. Right: transfer time only.

it is deployed, i.e., NDN and TCP/IP respectively.

Figure 2 shows the performance of ANDāNA and Tor with respect to their baselines. The graph on the left shows the measurements including the time required to setup a Tor circuit and all ephemeral circuits for ANDāNA. Session-based ANDāNA is denoted by ANDāNA-S, while ANDāNA with asymmetric encryption is referred to as ANDāNA-A. For small- to medium-size files (1-10MB), overhead of ANDāNA-A is between 1.5× and 1.75×. As expected, ANDāNA-S exhibits lower overhead (1.45× to 1.7×) due to more efficient symmetric encryption.

In comparison, Tor's download time for the same amount of data is between 2.3 and 7 times higher than that of TCP/IP. This imposes significant overhead for content size that fits many typical web pages. Whereas, ANDāNA is efficient in anonymizing such traffic patterns. Large file transfers are more efficient with Tor, which increases the total download time by about 1.4 times, compared to 2.4 and 2.1 of ANDāNA-A and ANDāNA-S.

The right-side graph in Figure 2 shows the relative speed of three approaches without including circuit setup time. Our measurements show that overhead of ephemeral circuit creation in ANDāNA-S is negligible. Since a new ephemeral circuit must be selected for every interest with ANDāNA-A, we simply report the same values from the previous graph. Results confirm that overhead of circuit creation in Tor is significant when retrieving small-size content. Removing this initialization phase from the measurements significantly reduces Tor's overhead. However, the overhead of ANDāNA with respect to its baseline is still smaller than that of Tor for content up to 10MB.

In absolute terms (comparing raw download times), Tor + TCP/IP performs better than ANDāNA + NDN in our testbed experiments. However, we believe that, in a realistic geographically-distributed deployment setting with limited-bandwidth links, ANDāNA + NDN would provide a significant performance advantage over Tor + TCP/IP due to its shorter (ephemeral) circuits. In other words, we anticipate that shorter circuits and content caching in ANDāNA + NDN would result in appreciably lower overall download times than Tor + TCP/IP in a global internet setting.

7 Conclusions and Future Work

Content-centric networking is a major transition from today's world that focuses on communication endpoints. NDN project represents one of the most visible current research efforts aiming to bring content-centric networking into the foreground by using it as a possible future Internet architecture. Despite some privacy-friendly features and side-effects, NDN poses some interesting privacy challenges. This work presents an initial attempt to provide anonymity in NDN. The main contribution of this work is threefold: (1) exploration of privacy issues in NDN, (2) design of an anonymization tool – ANDāNA, and (3) its security analysis and performance assessment.

At the same time, particularly because the entire NDN project (and, of course, ANDāNA) represent work-in-progress, one of the main goals of this paper is to solicit comments from the security research community. Also, since our work merely scratches the surface of privacy issues in content-centric networking and NDN, a number of issues are left for future work, including:

- More performance experimentation with ANDāNA, especially, in larger testbeds and under various traffic load / congestion scenarios. (This should lead to better code profiling and lower overhead.)
- Comprehensive directory service for effective large-scale distribution of up-to-date AR information.
- In-depth study of both privacy and performance trade-offs in the use of asymmetric vs. symmetric ANDāNA variants.
- DoS mitigation measures, such as computational puzzles for circuit establishment.
- Red-teaming experiments to assess realistic privacy attainable with ANDāNA.
- Modification of ANDāNA to support other emerging content-centric architectures and comparative experiments among them.

Acknowledgments

We thank NDSS'12 anonymous reviewers for their valuable feedback. We are also grateful to Van Jacobson, Jim Thornton, Kasper Rasmussen, Yanbin Lu, Lixia Zhang and Mark Baugher for their helpful input and comments on earlier drafts of this paper. This work was conducted in the context of the NSF project: "CNS-1040802: FIA: Collaborative Research: Named Data Networking (NDN)".

References

[1] M. Abdalla, M. Bellare, and G. Neven. Robust encryption. In *Theory of Cryptography Conference, TCC 2010*, 2010.

[2] A. Abdul-Rahman. The PGP Trust Model, 1997.

[3] Anonymizer anonymous surfing. http://www.anonymizer.com/.

[4] S. Arianfar, T. Koponen, S. Shenker, and B. Raghavan. On preserving privacy in content-oriented networks. In *ACM SIGCOMM Workshop on Information-Centric Networking*, 2011.

[5] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker. Low-resource routing attacks against anonymous systems. In *The 2007 Workshop on Privacy in the Electronic Society*, 2007.

[6] M. Bellare, B. A., D. A., and D. Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT*, 2001.

[7] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *CRYPTO*, 1996.

[8] M. Bellare and S. Miner. A forward-secure digital signature scheme. In *CRYPTO*, 1999.

[9] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology*, 21(4), 2008.

[10] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *EUROCRYPT*, 1994.

[11] T. Callahan, M. Allman, and V. Paxson. A longitudinal view of http traffic. In *The 11th international conference on passive and active measurement*, 2010.

[12] Content centric networking (CCNx) project. http://www.ccnx.org.

[13] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), 1981.

[14] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10), 1985.

[15] J. Daemen and V. Rijmen. The design of Rijndael: AES - the advanced encryption standard. Springer, 2002.

[16] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a type III anonymous remailer protocol. In *The 2003 IEEE Symposium on Security and Privacy*, 2003.

[17] W. Diffie and M. Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6), 1976.

[18] R. Dingledine, N. Mathewsonn, and P. Syverson. Tor: The second-generation onion router. In *The 13th USENIX Security Symposium*, 2004.

[19] J. Feigenbaum, A. Johnson, and P. Syverson. A model of onion routing with provable anonymity. In *Financial Cryptography*, 2007.

[20] National science foundation (NSF) future of internet architecture (FIA) program. `http://www.nets-fia.net/`.

[21] M. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *The 9th ACM Conference on Computer and Communications Security*, 2002.

[22] E. Gabber, P. Gibbons, D. Kristol, Y. Matias, and A. Mayer. Consistent, yet anonymous, web access with lpwa. *Communications of the ACM*, 42(2), 1999.

[23] M. Gritter and D. Cheriton. An architecture for content routing support in the internet. In *USENIX Symposium on Internet Technologies and Systems*. USENIX Association, 2001.

[24] C. Gülcü and G. Tsudik. Mixing e-mail with Babel. In *Network and Distributed Security Symposium*, 1996.

[25] A. Houmansadr, G. Ngyuen, M. Caesar, and N. Borisov. Cirripede: Circumvention infrastructure using router redirection with plausible deniability. In *The 18th ACM Conference on Computer and Communications Security*, 2011.

[26] I2P anonymous networking project. `http://www.i2p2.de/`.

[27] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard. Networking named content. *The 5th international conference on Emerging networking experiments and technologies*, 2009.

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

[29] P. Mittal, A. Khurshid, J. Juen, M. Caesar, and N. Borisov. Stealthy traffic analysis of low-latency anonymous communication using throughput fingerprinting. In *The 18th ACM Conference on Computer and communications security*, 2011.

[30] U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman. Mixmaster protocol — Version 2. IETF Internet Draft, 2003.

[31] S. Murdoch and R. Watson. Metrics for security and performance in low-latency anonymity systems. In *Privacy Enhancing Technologies Worshop*, 2008.

[32] Named data networking project (NDN). `http://named-data.org`.

[33] NetInf: networkd of information project. `http://www.netinf.org/`.

[34] Open network lab. `http://onl.wustl.edu`.

[35] L. Overlier and P. Syverson. Locating hidden servers. In *IEEE Symposium on Security and Privacy*. IEEE, 2006.

[36] PURSUIT - a fp7 european union project -. `http://www.fp7-pursuit.eu/PursuitWeb/`.

[37] M. Reiter and A. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), 1998.

[38] M. Rennhard and B. Plattner. Introducing morphmix: Peer-to-peer based anonymous internet usage with collusion detection. In *Workshop on Privacy in the Electronic Society*, Washington, DC, USA, 2002.

[39] M. Rennhard and B. Plattner. Practical anonymity for the masses with morphmix. In *Financial Cryptography*, 2004.

[40] A. Serjantov and P. Sewell. Passive attack analysis for connection-based anonymity systems. *European Symposium on Research in Computer Security*, 2003.

[41] V. Shmatikov and M. Wang. Timing analysis in low-latency mix networks: Attacks and defenses. *European Symposium on Research in Computer Security*, 2006.

[42] The OpenSSL Project. OpenSSL: The open source toolkit for SSL/TLS. `www.openssl.org`.

[43] M. Wright, M. Adler, B. Levine, and C. Shields. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Transactions on Information and System Security (TISSEC)*, 7(4), 2004.

[44] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman. Telex: Anticensorship in the network infrastructure. In *The 20th USENIX Security Symposium*, 2011.

A Security Proofs

Justification of Claim 5.1: Suppose that Claim 5.1 is false. Then, Adv can be used to construct an algorithm Sim that breaks the CPA-secure encryption scheme $\mathcal{E}$ as follows: Sim plays the CPA-security game with a challenger, that selects a public key pk. Sim selects a public key pk_2 and initializes Adv, that eventually returns two interests int^0, int^1 of its choice. Sim sends $c_0 = \mathcal{E}_{pk_2}(\mathsf{int}^0)$ and $c_1 = \mathcal{E}_{pk_2}(\mathsf{int}^1)$ to the challenger, that returns $c^* = \mathcal{E}_{pk}(c_b) = \mathcal{E}_{pk}(\mathcal{E}_{pk_2}(\mathsf{int}^b))$. Sim sends (c^*, c_0, c_1) to the challenger that eventually returns its choice b'. Sim outputs b' as its choice. The output of Sim is $b' = b$ iff Adv guesses b' correctly. Since Adv guesses b' correctly with non negligible advantage over $1/2$, Sim breaks the CPA-security of $\mathcal{E}$ with non negligible advantage. This violates the hypothesis of Claim 5.1, and, therefore, such Adv cannot exist. □

Proof of Theorem 5.1 — Consumer Anonymity (sketch). We prove that each condition in Theorem 5.1 implies consumer anonymity:

1. Assume that, for each $u' \neq u$ there exists no configuration $C' \equiv_{Adv} C$ with respect to Adv such that $C'(u') = C(u)$. Adv cannot determine that $C(u) \notin C'$ using only $C_2(u)$, $C_3(u)$ and $C_4(u)$: if $C_1(u) = C'_1(u')$ for some $C' \equiv_{Adv} C$ and u' (i.e. there exist an indistinguishable configuration with respect to Adv where a consumer different from u sends an interest to $C_1(u)$ through interface $\mathsf{if}_i^{C_1(u)}$ and $u, u' \in A_{\mathsf{if}_i^{C_1(u)}}$), then there must exist a tuple $C'(u') = C(u)$ since (a possibly compromised) r cannot process interests coming from consumers in the same anonymity set differently – that would imply that they are not in the same anonymity set. Therefore, for each configuration $C' \equiv_{Adv} C$, and for each $u' \neq u$ $\exists C'_1(u') = C_1(u) \Rightarrow \exists C'(u') = C(u)$.

   For this reason, $C'_1(u') \neq C_1(u)$ for all $C' \equiv_{Adv} C$ and for all $u' \neq u$, i.e. $\forall C'_1(u') = C_1(u).C(u) \notin C'$. This is true if and only if Adv controls at least one interface $\mathsf{if}_i^r \in \mathsf{path}^{C_4(u)}$ for which u' is not in the anonymity set of if_i^r, i.e., $\exists \mathsf{if}_i^r \in \mathsf{path}^{C_4(u)} \cap \mathsf{IF}_{Adv}$ s.t. $u' \notin A_{\mathsf{if}_i^r}$ Since this contradicts the hypothesis, there must exist a configuration C' indistinguishable from C with respect to Adv such that $C'(u') = C(u)$.
2. We assume that, for each $u' \neq u$, Adv can distinguish between interests from u from those from u' (i.e., condition 1 of theorem 5.1 does not hold). We show how to prove theorem 5.1 by reduction. Assume that there exists an efficient adversary Adv such that $\mathsf{C}_{Adv} = \mathsf{C} \setminus \{u, u'\}$ and $\mathsf{R}_{Adv} = \mathsf{R} \setminus \{r_1\}$ (i.e., Adv compromised all entities, except u, u' and r_1). Suppose that $C(u) = (r_1, r_2, p, \mathsf{int}^0_{pk_1,pk_2})$, $C(u') = (r_1, r'_2, p', \mathsf{int}^1_{pk_1,pk'_2})$ for some $r_2, r'_2, p, p', \mathsf{int}^0, \mathsf{int}^1$. For each C', Adv outputs: 1 on input of C and 0 on input of C' with non-negligible probability, where $C'(u) = C(u')$ and $C'(u') = C(u)$. In other words, there is no configuration for which $C \equiv_{Adv} C'$ holds. We sketch how Adv can be used as a subroutine in a simulator Sim that breaks Claim 5.1.

   Sim creates a random network topology N and inputs it to Adv. Sim also inputs the information that Adv would obtain by compromising all entities in N except u, u' and r_1. As such, Sim also includes $\mathsf{int}^b_{pk_1,pk_2}$ and $\mathsf{int}^0_{pk_2}$, $\mathsf{int}^1_{pk_2}$ received from the challenger of Claim 5.1 to the input of Adv. Then, Sim sends to Adv configurations C and C', where C is identical to C', except that $C(u) = C'(u')$ and $C(u') = C'(u)$, and $C(u) \neq C(u')$. We have that $b = 1$ iff Adv outputs 1. Since existence of Sim violates Claim 5.1, Adv cannot exits.
3. We assume that, for each $u' \neq u$, Adv can distinguish between interests from u from those from u' (i.e., condition 1 of theorem 5.1 does not hold) and that the first router in u's and u''s paths is compromised, i.e., condition 2 of theorem 5.1 does not hold. We then prove theorem 5.1 by reduction. Assume that there exists an efficient adversary Adv such that $\mathsf{C}_{Adv} = \mathsf{C} \setminus \{u, u'\}$ and $\mathsf{R}_{Adv} = \mathsf{R} \setminus \{r_2\}$ (i.e., Adv compromised all entities, except u, u' and r_2). Suppose that $C(u) = (r_1, r_2, p, \mathsf{int}^0_{pk_1,pk_2})$, $C(u') = (r'_1, r_2, p', \mathsf{int}^1_{pk'_1,pk_2})$ for some $r_1, r'_1, p, p', \mathsf{int}^0, \mathsf{int}^1$. For each C', Adv outputs 1 on input of C, and 0 on input of C', where $C'(u) = C(u')$ and $C'(u') = C(u)$. In other words, there is no configuration where $C \equiv_{Adv} C'$ holds. We sketch how Adv can be used as a subroutine in a simulator Sim to determine, given int_{pk_2} and int'_{pk_2}, whether $\mathsf{int} = \mathsf{int}'$.

   Sim creates a random network topology N and inputs it to Adv. Sim also inputs the information that Adv would obtain by compromising all entities in N except for u, u' and r_2. Sim interacts with the challenger of Claim 5.1 setting the innermost key of its challenge, denoted as $\overline{pk_2}$, to $\perp$. Sim receives $\mathsf{int}^b_{pk_1}$ for some $\mathsf{int}^0, \mathsf{int}^1$ of its choice, and adds $\mathsf{int}^b_{pk_1,\overline{pk_2}}$, $\mathsf{int}^b_{\overline{pk_2}}$ and $\mathsf{int}^b_{\overline{pk_2}}$ to the input of Adv. Then Sim sends to Adv configurations C and C', where C is identical to C' except that $C(u) = C'(u')$ and $C(u') = C'(u)$, and $C(u) \neq C(u')$. We have that $b = 1$ iff Adv outputs 1. Since the existence of Sim would violate Claim 5.1, Adv cannot exits. □

Proof of Theorem 5.2 — Producer Anonymity (sketch). We prove that each condition in Theorem 5.2 implies producer anonymity:

1. Let $C_4(u') = \mathsf{int}'_{pk_1,pk'_2}$ and let C' be identical to C except that $C'(u) = (C_1(u), C_2(u), C_3(u), C_4(u'))$ and $C'(u') = (C_1(u'), C_2(u'), C_3(u'), C_4(u))$. In other words, C' is a configuration where int_{pk_1,pk_2} is sent to a producer different from p. In this setting, Adv can only distinguish C' and C by distinguishing $C'(u)$ and $C'(u')$. Claim 5.1 guarantees that Adv that observes int_{pk_1,pk_2} and $\mathsf{int}'_{pk_1,pk'_2}$ cannot determine which corresponds to int and which – to int'. Moreover, Assumption 5.1 prevents Adv from linking the output of non-compromised router $C_1(u)$ with int_{pk_1,pk_2} and $\mathsf{int}'_{pk_1,pk'_2}$. Therefore, $C \equiv_{Adv} C'$.
2. Similarly, let $C_4(u') = \mathsf{int}'_{pk_1,pk'_2}$ and let C' be identical to C except that $C'(u) = (C_1(u), C_2(u), C_3(u), C_4(u'))$ and $C'(u') = (C_1(u'), C_2(u'), C_3(u'), C_4(u))$. We assume that $C_1(u)$ and $C_1(u')$ are compromised. In this setting, Adv can only distinguish between C' and C by distinguishing $C'(u)$ and $C'(u')$. Claim 5.1 guarantees that any Adv that observes int_{pk_1,pk_2} and $\mathsf{int}'_{pk_1,pk'_2}$ cannot determine which corresponds to int and which – to int'. Moreover, Assumption 5.1 prevents Adv from linking the output of non-compromised router $C_2(u)$ with int_{pk_2} and $\mathsf{int}'_{pk'_2}$. Therefore, $C \equiv_{Adv} C'$. □

B Performance Evaluation: Additional Results

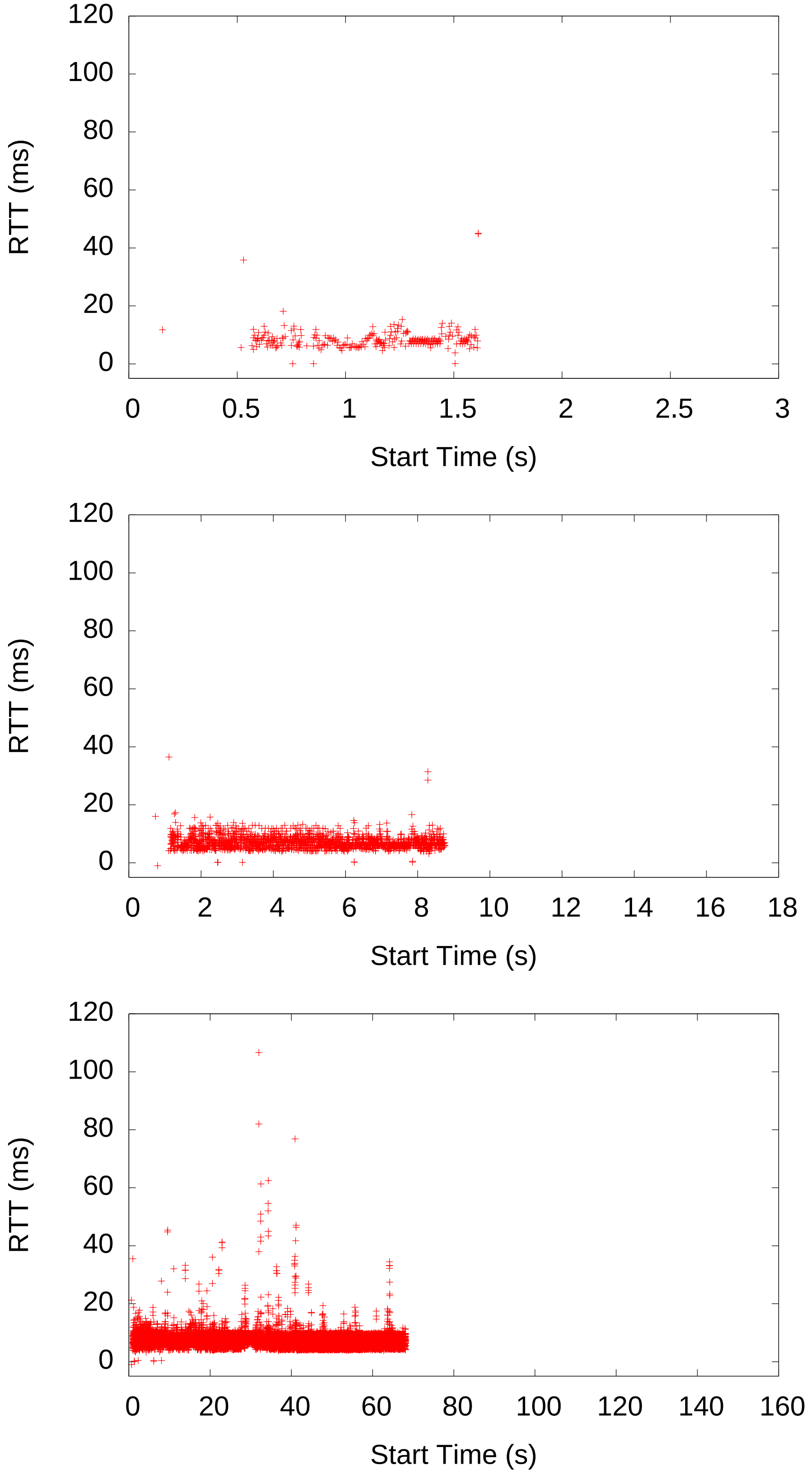

Figure 3. Round trip time for transferring 1, 10 and 100MB of content over NDN (limited anonymity)

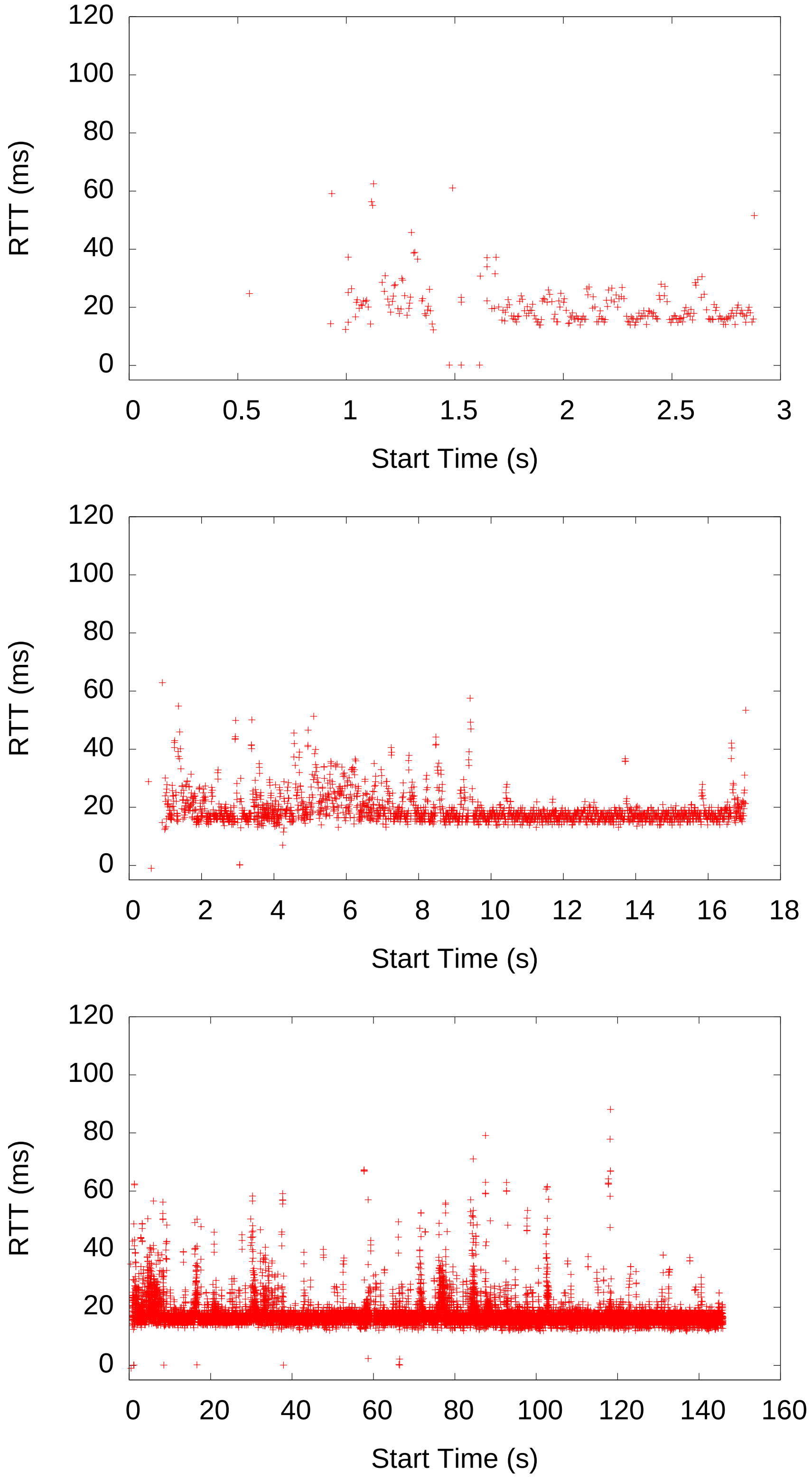

Figure 4. Round trip time for transferring 1, 10 and 100MB of content over ANDāNA (full anonymity).