Realizing a Virtual Private Network using Named Data Networking

Craig Partridge
Raytheon BBN Technologies
Cambridge, Massachusetts, USA
craig.partridge@raytheon.com

Samuel Nelson
Raytheon BBN Technologies
Cambridge, Massachusetts, USA
samuel.nelson@raytheon.com

Derrick Kong
Raytheon BBN Technologies
Cambridge, Massachusetts, USA
derrick.kong@raytheon.com

ABSTRACT

An approach to creating secure virtual private networks for the Named Data Networking (NDN) protocol suite is described. It encrypts and encapsulates NDN packets from higher security domains and places them as the payload in unencrypted NDN packets, much as IPsec encapsulates encrypted IP datagrams in unencrypted IP datagrams. We then leverage the well-known properties of the IP-in-IP approach, taken by IPsec in tunnel mode, to understand the strengths and weaknesses of the proposed NDN-in-NDN approach.

CCS CONCEPTS

• **Security and privacy → Security protocols**; • **Networks →** Network protocol design;

KEYWORDS

Named Data Networking, IPsec, VPN

ACM Reference Format:
Craig Partridge, Samuel Nelson, and Derrick Kong. 2017. Realizing a Virtual Private Network using Named Data Networking. In *Proceedings of ICN '17, Berlin, Germany, September 26–28, 2017,* 7 pages.
https://doi.org/10.1145/3125719.3125720

1 INTRODUCTION

Named data networking (NDN) is an innovative content networking protocol suite. To date, most work on NDN security has been focused on issues of authentication and correct operation of the NDN system [16]. In this paper, we tackle a different problem, namely whether we can design an NDN Virtual Private Network (VPN).

Specifically we look at a classic security challenge. Two or more secure *red* networks wish to exchange data, but are only connected via a less secure *black* network. The red networks need to somehow protect their traffic as it transits the black network.

Furthermore, we examine this problem using a classic approach. We seek to adapt a well-understood solution from another networking architecture, in this case, the Internet protocol suite. Adaptation has multiple intellectual advantages. It builds on a well-understood and well-analyzed approach. It can leverage assessment mechanisms developed for the existing approach. And it can highlight differences. This last point is important. Different protocol architectures have different strengths and weaknesses and applying ideas from one architecture to another is a well-known way to explore those strengths and weaknesses. Prior works in IPv6 security and NDN anonymity have used this approach with success [15][5].

Accordingly in this paper we seek to design a VPN architecture that encapsulates NDN inside NDN (NDN-in-NDN), just as the Internet architecture achieves VPNs using IP-in-IP. Note that to strengthen the challenge (and the insight) we assume NDN is the layer 3 protocol (vs. NDN solutions that run NDN over UDP and IP). The goal is to achieve a secure NDN VPN analogous to the IP-in-IP IPsec tunnel mode service offered by today's Internet-based VPNs and to do a straightforward comparative security analysis vis-à-vis the well-understood IP-in-IP approach.

We first describe the NDN-in-NDN approach, compare it to IP-in-IP, and discuss some interesting points of difference including replay protection, covert channels, NDN name obfuscation, and basic traffic analysis. We note that while there are other approaches to establishing NDN VPNs (some of which we discuss at the end of the paper), starting from the IPsec approach provides a tried-and-true framework for analyzing the security properties of NDN.

The analysis here is focused on attacks from outsiders. Mitigating attacks from entities inside secure networks and from compromised secure nodes is left for a later study.

2 IP VIRTUAL PRIVATE NETWORKS

The Internet's IP security model (IPsec) has been in use for over 20 years and was the culmination of over 20 years of research into securing packet based data networks above the link layer starting with the ARPANET private line interface [6]. In its current incarnation, IPsec is a well-understood and well-grounded security architecture, and is the foundation behind modern IP VPN service [9].

The basic notion is that there are two or more secure (by convention *red*) networks that wish to exchange IP datagrams and must do so over an unsecured (*black*) IP network, often the Internet. The red networks are attached to the black network via *gateways.* To get a red datagram from one red network to the other, the source network's gateway encrypts the red datagram and places the encrypted datagram as the payload of an unencrypted datagram. It then transmits this IP-in-IP datagram over the black network to the destination red network's gateway, where the encrypted payload

This document does not contain technology or Technical Data controlled under either the U.S. International Traffic in Arms Regulation or the U.S. Export Administration Regulations.

ICN '17, September 26–28, 2017, Berlin, Germany

ACM ISBN 978-1-4503-5122-5/17/09…$15.00
https://doi.org/10.1145/3125719.3125720

is decrypted and forwarded into the destination red network. The degenerate case in which the source or destination red network is a single host is supported without requiring special handling. An important feature of the IPsec solution is that the IP service on the black (and red) networks does not have to change to support security. We seek the same result for NDN.

Because IPsec VPNs are a mature technology there are a number of security profiles and requirements to help implementations realize security operation. We have used one such profile, the NIST SP 800-77 requirements [12], shown in Table 1.

3 OVERVIEW OF NAMED DATA NETWORKING (NDN)

At its core, NDN has three innovations. First, it uses the publish-subscribe (pub-sub) data communications model as its network layer protocol architecture. Second, unlike many pub-sub models where the publisher and subscriber are tightly linked (e.g. DDS [14]), NDN disassociates the publisher and subscriber. In NDN the publisher publishes into the *network*, and the subscriber requests data from the *network*. At no time do the publisher and subscriber need to be in direct communication; in fact, heavy use of in-network caching promotes the serving of content from nodes that were not the original publisher. Third, NDN vests all the responsibility for reliable data retrieval in the subscriber.

The result is a simple protocol in which a subscriber sends an *Interest* packet requesting data and the network responds with a *Data* packet, containing data that may have been previously published or newly generated by the publisher. To ensure that cached *Data* is an accurate copy of what the publisher produced, all *Data* packets are digitally signed. Each *Data* packet has a name – names are hierarchical and used directly in routing tables. When the network receives an *Interest* for unknown data, the information in the name is used to efficiently route the *Interest* towards a publisher who can serve it. *Interests* are aggregated so that only one copy of an *Interest* typically crosses a link and only one copy of a *Data* is returned over any link.

While NDN sounds simple, its architecture is very different from the sender-directed communications architectures primarily used today. Perhaps the simplest example of the difference is that, in NDN, there's no direct way to "send" – to transmit unsolicited data to another party. As a result, application protocols look different in NDN.

4 NDN-IN-NDN

In this section, we describe a simple NDN VPN by encapsulating red NDN packets in black NDN packets, just as IP VPNs encapsulate red IP datagrams in black IP datagrams. The NDN services on both red and black networks are the same. We call this approach *NDN-in-NDN*.

While there are other ways to create an NDN VPN, and we discuss some in Section 7, because NDN-in-NDN mirrors the approach of IP VPNs, it is easier to compare NDN-in-NDN and IP VPNs. Another advantage is that encapsulation allows us to equally easily connect red NDN networks or red NDN hosts via the black NDN network.

The goal of NDN-in-NDN is to provide the full set of NDN features to the red NDN nodes, even when their data is transiting the black network. Specifically, NDN caching and Interest aggregation must work seamlessly.

This section starts by describing a trivial NDN-in-NDN encapsulation, to set context. At the end of the section we observe that this simple NDN-in-NDN has several issues and we propose fixes for those issues before moving to evaluating the security of NDN-in-NDN in the following section.

4.1 Simple NDN-in-NDN Data Flow

Figure 1 illustrates how data flows through an NDN-in-NDN system. A host Alice issues an NDN Interest for the Data named */bbn.com/videos/v1.mpg* in its red enclave. The Data to fulfill this interest is in another red enclave, so the Interest is routed to a gateway. At the gateway, subject to security policies (e.g. Requests for */bbn.com/videos/* may be prohibited from leaving the red enclave), the red Interest is converted to a black Interest and placed on the black network. Note that in the simplest case, the black-side name is the same as the red-side name. This key issue with the simple NDN-in-NDN approach is examined in Section 4.2.

The black Interest is then routed using Forwarding Information Base (FIB) entries to a red NDN enclave that contains a publisher, Bob, for */bbn.com/videos/v1.mpg* (recall there may be multiple publishers for the same name). The gateway at the publisher's enclave translates the black Interest into a red Interest and transmits the red Interest to Bob.

Bob responds with a Data packet for */bbn.com/videos/v1.mpg*. The Data packet follows the reverse data path of the Interest packet to the publishing enclave's gateway, where the information in the Data packet is encrypted with a shared red domain key and then a black Data packet for */bbn.com/videos/v1.mpg* with the encrypted red-side contents is placed on the black network. This black Data packet follows the reverse path in the black network to the gateway for Alice's enclave, where the black Data packet's content is decrypted using the shared red key and converted to a red Data packet for */bbn.com/videos/v1.mpg* and forwarded along the reverse path in the red enclave to Alice.

What if a second red requester Charlie, also wants to see */bbn.com/videos/v1.mpg*? All the NDN features work. Charlie creates the red side Interest. If Charlie and Alice are in the same red enclave, a red cache in their shared enclave will satisfy Charlie's Interest. If not, Charlie's Interest moves to the black side network, where it will be filled by a copy of */bbn.com/videos/v1.mpg* from a black side cache (possibly the cache at the Bob's gateway) and returned to Charlie.

Simple NDN-in-NDN has a highly desirable feature: *it preserves all of NDN's features.* Interests work across the red and black networks. Caching works – a red Data packet can be cached (in encrypted form) on the black network and can be used to satisfy requests from red network consumers. Interests from multiple red network consumers can be consolidated on the black side into a single Interest to the red side publisher. Within each domain (black and red) simple NDN-in-NDN has exactly the same semantics as regular NDN. All we have done is split those domains.

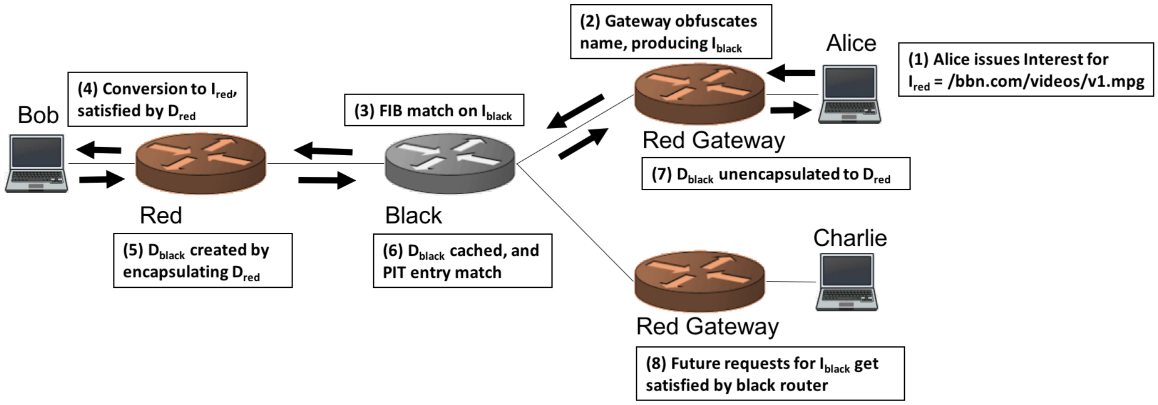
(2) Gateway obfuscates name, producing I_{black}
Alice
(1) Alice issues Interest for I_{red} = /bbn.com/videos/v1.mpg
(4) Conversion to I_{red}, satisfied by D_{red}
Bob
(3) FIB match on I_{black}
Red Gateway
(7) D_{black} unencapsulated to D_{red}
Red
Black
(5) D_{black} created by encapsulating D_{red}
(6) D_{black} cached, and PIT entry match
Charlie
Red Gateway
(8) Future requests for I_{black} get satisfied by black router

Figure 1: Interest and Data flow through NDN-in-NDN system, providing a virtual private network

4.2 Securing Simple NDN-in-NDN

The simple NDN-in-NDN approach described in Section 4.1 has several security issues, which we now address.

4.2.1 Black to Red Interest Translation is Not Secure. Black Interests for names are simply placed in the red enclave. The red enclave has no way to confirm that the Interest actually originated at a red host. As a result, a black side system can perform a denial-of-service attack on the red enclave with Interests that seek to fill the red PIT with random requests, or the red data caches with unsolicited data. Furthermore, there is an opportunity for a black side node to use names in Interests to convey information into the red side.

The obvious way to protect the red enclave fix is to place some amount of authenticating material from the red requesting host in the black Interest. One possibility is to encrypt the red Interest and carry it in the black Interest. Another is simply to carry the red authenticating information (again, encrypted). This information can be checked by the publisher's gateway to ensure the black Interest reflects a valid red Interest. (NDN has recently been extended to support signed Interests [1]).

Note, however, that this still permits potential denial-of-service attacks on the publisher's gateway using bogus Interests that the gateway must authenticate. We have not identified a solution to this problem.

4.2.2 Information about the Red Enclaves Leaks in Names. Observe that in the simplest case NDN-in-NDN repeats the red *Name* on the black side. While this makes the NDN mechanics simple, it means that information about the structure of the data and its publishers in red enclaves is being transmitted on the black network.

One solution is to obfuscate names on the black side. The obfuscation mechanism needs to be consistent (all red networks need to translate names in the same way) and injective – the Interest and Data *Name* on the red size, $Name_{red}$ must translate to the same black name, $Name_{black}$, for Alice and Bob and Charlie, and two different red names may not translate to the same black name, or forward and caching won't work on the black side. Also, there needs to be some way for the red enclaves to advertise the black names of their publishers into the black routing system.

Depending on the degree of privacy needed, this is a challenging problem, and one that has been extensively explored in past literature [7, 11]. We discuss this topic in further detail in Section 6.3.

4.2.3 Shared Red Domain Keys. Simple NDN-in-NDN requires that all the red enclaves have access to the same set of keys. Recall that a consumer sending an Interest may not know which red enclave contains the publisher. And due to Interests being fulfilled by Data in black caches, the publisher does not have direct interactions with all consumers. A shared set of keys ensures that all communications patterns work correctly. Pairwise keys, as are used in IPsec, do not work as pairwise keys inhibits black-side caching (Data encrypted with keys for Alice and Bob cannot be cached for later retrieval by Charlie).

While this approach may wear out a key faster, it is not a bug. Applications that want additional security can do either key agreement (red application to red application) and encrypt their application data, or use intra-domain group-based cryptographic techniques such as Attribute Based Encryption (ABE) [3]. This results in layered encryption approach, which is specifically called out as a positive feature in the NSA Commercial Solutions for Classified Mobile Access Capability Package [13].

Similar to IP VPNs, it is likely that all members of an NDN red domain will be logically part of a centralized, administrative domain (e.g., a company or university). This makes the establishment of group keys feasible and realistic. Furthermore, existing PKI techniques can be used to securely deliver group keys to each red member individually, effectively bootstrapping the system. This same PKI approach can be used to remove a member from the red domain, simply by sending all non-revoked members updated group keys.

5 NDN-IN-NDN AND IP VPN SIDE BY SIDE

In this section, we consider the individual steps of creating and using an IP VPN using IPsec and examine the analogous steps in NDN-in-NDN. In particular, we examine how our NDN-in-NDN approach operates in relative to IP VPNs.

Overall, the side-by-side comparison reveals that, even though their underlying protocols are very different, IP-in-IP and NDN-in-NDN are often quite similar. They encapsulate packets similarly.

Table 1: How IP-in-IP and NDN-in-NDN meets NIST SP 800-77 requirements

NIST SP 800-77 requirements	IP-in-IP	NDN-in-NDN
Confidentiality protection	Encrypt red IP datagram in black IP datagram	Encrypted red NDN packet in black NDN packet
Integrity protection	AH header	Use NDN built-in integrity checks
Replay protection	ESP sequence numbers	Authenticated Interests; NDN Data needs no replay protection
Security association lifetimes	Rekey security associations	Rotate domain-specific keys

Security gateways must perform certain consistency checks. But there are also differences. NDN has two distinctly different packet types (Interest and Data), and NDN leverages in-network caching. We illustrate how both IP-in-IP and NDN-in-NDN meet the NIST SP 800-77 VPN requirements in Table 1.

5.1 Creating the Security Association

In an IPsec IP VPN, the central building block is a Security Association (SA). An SA is a unidirectional relationship between two endpoints (e.g. security gateways) that allows one endpoint to securely send packets to the other endpoint.

To connect two red IP subnetworks over a black IP network one must create two SAs (one each direction) between security gateways on each network. There is a SA negotiation protocol, the Internet Security and Key Management Protocol (ISAKMP), to establish security associations between two gateways. Each SA has its own set of encryption keys and a set of policies that define the traffic that may use the SA. To connect three red IP subnetworks, one must create pairwise connections between all the networks.

In NDN-in-NDN there is a single SA for all participants in the red domain. All gateways in that domain use a shared set of keys to encrypt and decrypt red NDN packets encapsulated in black NDN packets. Any gateway can send to any other gateway within that domain.

NDN-in-NDN's SA is different because its data delivery service is different. In an IP network, a datagram has a destination IP address, so in an IPsec world, the outbound gateway can use the destination address to determine the SA on which to place the datagram. But in NDN, if the red packet is an Interest, the gateway has no idea which of its peer red subnetworks contains the publisher for the requested data. All the gateway can do is place the Interest into the black NDN network and let the Interest flow to the right red network.

Note that each red subnetwork could encrypt using a unique key. NDN-in-NDN would still work correctly. But all the peer subnetworks would have to know how to decrypt packets from all the other peer subnetworks.

Note that selective key distribution, in which a red subnetwork only shares its key with some of the subnetworks, doesn't work. It creates the possibility that a black Interest will flow to a red subnetwork that is unable to decrypt the encapsulated red Interest. That may sound acceptable – perhaps the destination red subnetwork is one the originating subnetwork doesn't want to trust – but consider this scenario where a publisher is duplicated in two red subnetworks, one trusted and one not. There is no way to specify this trust model in the black Interest, so the black Interest could be delivered to the untrusted subnetwork (and discarded) when it could have been delivered and served by the trusted subnetwork.

5.2 Sending to Another Subnetwork

Bob wants to send an IP datagram to Alice. Alice wants to request a piece of data over NDN from another red subnetwork. How do these communications progress to the receiving red subnetwork?

In IPsec, Bob's datagram is addressed to Alice's red IP address. It will be routed from Bob's host to the gateway for Bob's red subnetwork. The gateway will look up Alice's red IP address in a table that will return the SA between Bob's red subnetwork and Alice's red subnetwork. If there is no SA between the subnetworks, Bob's datagram will be discarded. If there is an SA there may be SA-specific checks done on Bob's IP datagram. For instance, the SA may only permit some protocols through (e.g. TCP and UDP but not ICMP). If Bob's IP datagram is acceptable, then an integrity check is calculated and the datagram is encrypted and placed inside an Encapsulating Security Payload (ESP) datagram, which identifies the security association and also includes a sequence number. The ESP datagram is then placed within a black IP datagram addressed to the gateway for Alice's subnetwork and transmitted over the black network.

In NDN, Alice issues an Interest for the name of the data Bob is sharing, $DataName_{red}$. The Interest moves through Alice's red subnetwork until it reaches the gateway. Just like the IPsec gateway looked up the destination IP address, the NDN gateway looks up $DataName_{red}$ to determine if it should put the Interest onto the black NDN network. It may be, for instance, that the publisher for $DataName_{red}$ is in Alice's subnetwork and the NDN gateway has a rule against forwarding the Interest to the black NDN network. Assuming the Interest should be forwarded to the black side, the gateway will add an integrity check to the Interest (if it does not already have one), encrypt the Interest, add a domain identifier (akin to an SA identifier), generate the obscured black NDN name $DataName_{black}$, add some form of replay protection (see Section 6.1) and issue the black Interest for $DataName_{black}$ with the encrypted red Interest as additional data in the black Interest. The black Interest is routed by the black NDN network towards the gateway of a publisher.

5.3 Receiving from Another Subnetwork

Bob's datagram arrives at the IPsec gateway for Alice's red subnetwork. Alice's Interest for $DataName_{black}$ arrives at a gateway. What happens next?

Bob's datagram has arrived at the receiving end of the SA. The IPsec security gateway will do a set of checks on the black and red version of the datagram including confirming the black datagram is not a replay of an older datagram, confirming that the datagram was encrypted and authenticated by an authorized user of the SA and that the header fields in the red datagram have values permitted in datagrams sent over the SA. If the datagram passes the checks it is forwarded on the red subnetwork to Alice's host.

The arrival of Alice's Interest gets a similar treatment. The receiving gateway decrypts the red Interest for $DataName_{red}$. It then must do a series of checks. It should confirm the decrypted NDN packet was indeed an Interest and not Data. The gateway also needs to confirm that $DataName_{black}$ was indeed an appropriate obfuscated name for $DataName_{red}$. Both checks ensure this is not an attempt to sneak an inappropriate NDN datagram past the gateway. If it passes the checks, the red Interest is forwarded into the subnet.

Note that NDN permits Interests to be multicast to multiple potential publisher nodes. So black Interest may be delivered to multiple gateways, only some of which actually have a publisher that can fulfill an Interest in $DataName_{red}$. When the Interest for $DataName_{red}$ is forwarded into the red subnetwork, it may promptly be dropped for lack of a publisher.

5.4 Sending Data or an ACK Back

Bob's IP datagram arrives at Alice's red host and Alice's host sends an acknowledgment (ACK). Alice's NDN Interest for $DataName_{red}$ arrives at a publisher, Bob, who has the Data named $DataName_{red}$ and Bob sends a Data packet back to Alice. How do these packets work their way through IPsec and NDN-in-NDN?

Alice's ACK has an experience just like Bob's inbound datagram. The ACK is routed to Alice's gateway, which places an encrypted datagram containing the ACK on an SA to Bob's red subnetwork. On receipt of the encrypted datagram, the gateway at Bob's subnetwork does the validity checks and decryption and sends the ACK to Bob's host.

Bob's NDN Data packet has a richer experience. Bob's Data for $DataName_{red}$ follows the reverse path of Alice's Interest back to the gateway for Bob's red subnetwork. It is also cached in the red NDN nodes it transits. At the gateway, the Data packet is encrypted and given a domain identifier and the obfuscated name $DataName_{black}$, the same name used in the Interest. The encrypted Data packet is then placed on the black network.

Note that unlike the Interest, there's no need to add authentication information, as NDN Data packets are required to be authenticated. Also, there's no need for replay protection as NDN encourages serving NDN Interests with Data from caches, which means a later Interest may be served with a copy of this black NDN Data packet.

The black NDN Data packet then follows the reverse path of the Interest to the gateway to Alice's red subnetwork. At each hop the black Data packet is cached.

When the black Data arrives at the gateway for Alice's subnet, the gateway performs a set of checks that ensure the decrypted red Data packet is consistent with the black Data packet (e.g. the obfuscated names are consistent and that the decrypted packet is indeed a Data packet). The red Data packet with $DataName_{red}$ then follows the Interests in the subdomain to Alice's host and delivery.

6 NDN-IN-NDN SECURITY TOPICS

Having sketched the mechanics of NDN-in-NDN, we now look at some of the security issues that were not fully addressed in Section 5. Each of these topics is rich with potential future research, and here we present a broad, surface-level treatment.

6.1 Replay Protection

A replay attack is a security attack in which a valid data transmission sent some time in the past is transmitted (usually by a third party) to cause harm to the network, end systems, or applications. Because there is such a wide range of possible replay attacks, IP-in-IP security solutions include mechanisms that swiftly treat previously transmitted packets (or overly delayed packets) as invalid.

However, the NDN architecture encourages the retransmission of valid data well after its original transmission. Specifically, NDN encourages caching and retransmission of Data packets, making Data packet replay attacks not harmful. Either a retransmitted Data packet is in response to an Interest, in which case it is valid, or it is not in response to an Interest, in which case all next hop recipients will discard it.

Observe that while retransmitting Data packets is useful, there are potential replay attacks using Interests. Consider what happens when a previously sent Interest is retransmitted some time later. The Interest will propagate through the network until either it is matched with Data in an intermediate cache or the Interest gets to a publisher who can (re)issue the Data. The replayed Interest will cause caches to be reloaded with the requested Data, which can drive other content out of the cache and harm network performance (e.g. the cached data driven out of the cache would have served a subsequent Interest). Affecting the cache is not a benign act. For instance, cache may be full, so reloading the old Data to match the replayed Interest may cause more useful Data to be discarded.

Note that a black Interest (encapsulating a red Interest) can be replayed by a black side node and affect caches in the publisher's red subnetwork.

A simple solution is to add replay protection and indeed, NDN has recently been extended to include replay protection for Interests - the authentication information added to Interests also includes a timestamp to bound the Interest's lifetime [1].

6.2 Bypass Channels

IP-in-IP security typically requires support for a bypass channel. As its name suggests, a bypass channel enables unencrypted/decrypted data to pass between the black and red networks. The primary reason is to dynamically set up SAs. A red subnetwork that wishes to attach to another red subnetwork needs to emit and receive specific black side packets to create a pair of security associations [8].

In NDN-in-NDN there is no need for a black to red bypass, since all red networks within a single red domain share group symmetric keys, and these keys can be exchanged using public key encryption over the NDN-in-NDN network, provided each domain has appropriate keying material (e.g. certificates from the same certificate authority). There is no situation in which the black side needs to pass Interests or Data directly to the red side, since all transfers of this nature will go through the standard procedure of checking and cryptographically transforming Data and Interest packets.

There is a need for the red side to tell the black side what obfuscated prefixes to advertise for FIB population. However, this channel follows similar standard NDN-in-NDN procedure for name obfuscation.

6.3 Name Privacy

NDN names are semantically meaningful by design, allowing efficient content dissemination. Forwarding on semantically meaningful names, however, poses a security problem. There are two competing desires – to preserve red side name privacy and be able to efficiently forward on names in the black side. Obfuscating names using encryption is a clear solution; however, there are numerous subtle challenges, many of which have been previously explored [7, 11]. For instance, any preservation of hierarchy may leak information, but routing is not scalable without it. Furthermore, if variable-length fields are not encrypted in a way that forces a fixed length, information can be leaked.

There is a tradeoff between privacy and scalability, and we argue that the situation and environment can help dictate the appropriate solution. When the structure of the hierarchical content names is not as important to protect, it makes sense to preserve some of the hierarchy on the black side to enable scalable routing. Exploring how much structure to collapse and under what circumstances to do it is a rich area for future research. Furthermore, advances in homomorphic encryption may be applicable to enable greater privacy. If black side nodes can build meaningful FIB and PIT tables from encrypted name announcements and Interests, and encrypted names could be matched against them in a homomorphic fashion, both privacy and scalability would be possible.

6.4 Traffic Analysis

A black side observer monitoring traffic both passing through or radiating from a node can infer certain red side actions in both IP-in-IP and NDN-in-NDN. One interesting question is how the inferences differ.

In the IP VPN case, an observer or observers monitoring traffic could infer which specific nodes are talking to which specific servers, and at which times. For instance, it could be inferred that both Alice and Charlie are speaking with Bob at the same time. Further, the content can often be inferred by examining the timing of packets in the stream [4].

In the NDN VPN case, the observer infers access to *content* not *end hosts*. For instance, it could be determined from the black side Data names that Alice and Charlie both accessed the same content (although the content itself would be unknown) even at *different* times. However, unlike the IP VPN case, it is quite possible, given caches, a single observer cannot infer exactly who produced that content.

7 OTHER NDN-IN-NDN STYLE APPROACHES

Other efforts have sought to create an NDN-style network with VPN-like qualities. We summarize them here.

The most similar work is CCNxKE [10]. CCNxKE encapsulates its content in a manner similar to NDN-in-NDN and uses encrypted names. However, the encrypted content are not cacheable as the encrypted names are not mapped to the unencrypted names. CCNxKE also is designed to implement communication between individual consumers (rather than enclaves) and publishers and incorporates a session protocol to this end.

ANDaNA [2] focuses on application-level protection and protection against traffic analysis via encrypted intermediate tunnels. Publishers encrypt their data using keys that are available only to authorized consumers. Features of this approach include that it intrinsically supports fine-grained access control (which NDN-in-NDN can only achieve by adding another layer of encryption/authentication) and allows NDN semantics throughout the network. However, it does not provide any boundaries between red and black networks and leaves obscuring of names to application discretion.

The different approaches of CCNxKE, NDN-in-NDN, and ANDaNA highlight a difference of opinion about who is in control of distribution (the publisher or the enclave), the importance of preserving NDN semantics (especially cacheability of content) in the black network, and how to deal with traffic analysis.

8 CONCLUSION

Comparing NDN-in-NDN with IP-in-IP raised some interesting security points. NDN-in-NDN can retain NDN semantics in the black network (red data can be cached and accessed) and does not require bypass channels in gateways. NDN-in-NDN does require that Interests be both authenticated and contain some form of replay protection.

We also want to repeat that this study was entirely focused on outsider attacks. Attacks by insiders within red subnetworks and attacks that leverage compromised red nodes or gateways are subjects that we leave for later study.

ACKNOWLEDGMENTS

We thank our shepherd, Jan Seedorf, and the anonymous reviewers for their comments, insights, and guidance.

REFERENCES

[1] 2017. Signed Interest. http://named-data.net/doc/ndn-cxx/current/tutorials/signed-interest.html. (2017).

[2] Alexander Afanasyev, J. Alex Halderman, Scott Ruoti, Kent Seamons, Yingdi Yu, Daniel Zappala, and Lixia Zhang. 2016. Content-based Security for the Web. In *Proceedings of the 2016 New Security Paradigms Workshop (NSPW '16)*. ACM, New York, NY, USA, 49–60. https://doi.org/10.1145/3011883.3011890

[3] John Bethencourt, Amit Sahai, and Brent Waters. 2007. Ciphertext-Policy Attribute-Based Encryption. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP '07)*. IEEE Computer Society, Washington, DC, USA, 321–334. https://doi.org/10.1109/SP.2007.11

[4] D. Cousins, C. Partridge, K. Bongiovanni, A. W. Jackson, R. Krishnan, T. Saxena, and W. T. Strayer. 2003. Understanding encrypted networks through signal and systems analysis of traffic timing. In *2003 IEEE Aerospace Conference Proceedings (Cat. No.03TH8652)*, Vol. 6. https://doi.org/10.1109/AERO.2003.1235227

[5] Steve DiBenedetto, Paolo Gasti, Gene Tsudik, and Ersin Uzun. 2011. ANDaNA: Anonymous Named Data Networking Application. *CoRR* abs/1112.2205 (2011). http://arxiv.org/abs/1112.2205

[6] Quinn DuPont and Bradley Fidler. 2016. Edge Cryptography and the Codevelopment of Computer Networks and Cybersecurity. *IEEE Ann. Hist. Comput.* 38, 4 (Oct. 2016), 55–73. https://doi.org/10.1109/MAHC.2016.49

[7] Cesar Ghali, Gene Tsudik, and Christopher A. Wood. 2016. (The Futility of) Data Privacy in Content-Centric Networking. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society (WPES '16)*. ACM, New York, NY, USA, 143–152. https://doi.org/10.1145/2994620.2994639

[8] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen. 2014. *Internet Key Exchange Protocol Version 2 (IKEv2)*. STD 79. RFC Editor. http://www.rfc-editor.

org/rfc/rfc7296.txt http://www.rfc-editor.org/rfc/rfc7296.txt.

[9] S. Kent and K. Seo. 2005. *Security Architecture for the Internet Protocol.* RFC 4301. RFC Editor. http://www.rfc-editor.org/rfc/rfc4301.txt http://www.rfc-editor.org/rfc/rfc4301.txt.

[10] Marc Moskoa, Erzin Uzun, and Christopher A. Wood. 2017. Mobile Sessions in Content-Centric Networks. In *16th Intl. IFIP TC6 Networking Conference, Networking 2017.*

[11] Edith Ngai, Borje Ohlman, Gene Tsudik, Ersin Uzun, Matthias Wahlisch, and Christopher A. Wood. 2017. Can We Make a Cake and Eat It Too? A Discussion of ICN Security and Privacy. *SIGCOMM Comput. Commun. Rev.* 47, 1 (Jan. 2017), 49–54. https://doi.org/10.1145/3041027.3041034

[12] NIST. 2005. *Guide to IPsec VPNs.* Technical Report.

[13] NSA. 2016. *Information Assurance Directorate Mobile Access Capability Package, Version 1.8.* Technical Report.

[14] Object Management Group. 2004. Data Distribution Service for Real-time Systems Specification. (December 2004). `http://www.omg.org/spec/DDS/1.0/PDF/`.

[15] C. Partridge, A. W. Arsenault, and S. T. Kent. 2007. Information Assurance and the Transition to IP Version 6 (IPv6). In *MILCOM 2007 - IEEE Military Communications Conference.* 1–8. https://doi.org/10.1109/MILCOM.2007.4455227

[16] Reza Tourani, Travis Mick, Satyajayant Misra, and Gaurav Panwar. 2016. Security, Privacy, and Access Control in Information-Centric Networking: A Survey. *CoRR* abs/1603.03409 (2016). http://arxiv.org/abs/1603.03409