# Blocking-Resilient Communications in Information-Centric Networks using Router Redirection

Hamid Mozaffari
*College of Information and Computer Sciences*
*University of Massachusetts*
Amherst, USA
hamid@cs.umass.edu

Amir Houmansadr
*College of Information and Computer Sciences*
*University of Massachusetts*
Amherst, USA
amir@cs.umass.edu

Arun Venkataramani
*College of Information and Computer Sciences*
*University of Massachusetts*
Amherst, USA
arun@cs.umass.edu

***Abstract*—Information-centric network (ICN) designs are susceptible to censorship especially packet filtering based on content names. Previous works on censorship circumvention in ICN either have high processing times or use proxies that can be blocked easily by the censoring agents. We design a new censorship circumvention approach for ICN using router redirection that enables a client in a censored region to retrieve blocked content from a censored destination without the censoring agent detecting the use of a censorship circumvention tool. We conduct ndnSIM-based simulation experiments showing that our approach is practical with only a modest end-to-end delay overhead.**

## I. Introduction

Today's Internet communication is carried over the TCP/IP protocol, whose host-to-host communication model has been widely criticized as misaligned for content retrieval. A multitude of next-generation network architectures, referred to as Information-Centric Networks (ICN), cache named content to improve overall network efficiency. ICN replaces host-to-host communication with an information-centric approach that retrieves named data regardless of the publisher's origin.

A major threat to ICN communications, similar in spirit to the current Internet, is *censorship* by repressive regimes and governments to prevent the open circulation of information. The major techniques used to deploy censorship in the current Internet are IP address blocking, DNS hijacking, and Deep Packet Inspection (DPI). Due to architectural differences, however, different mechanisms can be used to deploy censorship in ICNs; therefore, *existing techniques for bypassing censorship in the current Internet are insufficient to thwart censorship in ICNs*. Specifically, the following aspects of ICN communications amplify the censorship threat:

1) *Name leakage*: Unlike today's Internet, ICN packets carry the name of content inside themselves.
2) *Content leakage*: Encryption of content in ICN forces a trade-off between the effectiveness of in-network caching and leakage of the content being retrieved.
3) *Signature leakage*: Each packet carries its publisher ID, a public key and the signature of the content producer.

This work was supported by Intel and NSF under grant CNS-1719386.

Previous work has looked at designing circumvention techniques for ICNs by using cryptographic operations or proxies. For instance, Fotiou et al. [1] advocate homomorphic encryption that has a prohibitively high overhead. Tourani et al. [2] encode names with Huffman coding so that a proxy can decode the names. The major downside of reliance on a third-party proxy is that it can be blocked by the censors.

In this paper, we design a new circumvention protocol for ICN communication. The main advantage of our technique is that a client in a censored region can maintain unobservable communication with a censored destination, i.e., the censoring agent is oblivious to whether the client is using a censor circumvention tool. Our primary focus is on addressing name leakage, however, we propose approaches for other concerns too. Our circumvention method is inspired by routing-based circumvention techniques designed for the current Internet referred to as decoy routing [3]–[5].

Our protocol relies on the collaboration of some friendly routers in the uncensored portion of the network. Each client informs the redirecting routers about its packets in the registration phase. After registration, the client probes different destinations to find content such that the interest packets pass through these routers. Then the client signals to the router that its packets should be redirected to the real censored destination. When the redirecting router receives such interest packets, it retrieves the blocked content. We propose two approaches for this traffic redirection. First, we propose to use *ephemeral names* for the redirecting routers, such that only a client registered in the system knows the corresponding real names. Second, we propose to leverage one-on-one protocols in ICN, e.g., embeding our protocol in CCNxKE [6], a secure key exchanging protocol in ICN. The client can embed its messages inside the different fields of this protocol covertly, and request blocked content without attracting any attention from the censoring agent.

We simulate our censorship evasion approach inside ndnSIM [7], an NS3-based named data networking (NDN) simulator. We evaluate the performance of our approach for different file sizes and bottleneck bandwidths. The results

show that with a practical amount of additional delay (15%-50%), a client can retrieve blocked content in a censored destination in a manner such that the censor is oblivious to the client's use of a censorship circumvention tool.

**Summary of the contributions:** Our primary contribution is the design of a new censorship circumvention technique for ICNs that enables a client to use router redirection to circumvent the censoring agent and retrieve blocked content. To this end, our work makes the following technical contributions:

- Design of a scheme based on ephemeral names known only to the client;
- Design of a scheme to embed the protocol within CCNxKE, a pre-existing secure key exchange protocol;
- ndnSIM-based simulations showing the modest overhead of our protocol with varying file sizes and bandwidths;

## II. Problem Statement

We design a circumvention technique for the following problem: A client is located in a censored region (e.g., China), and all of its traffic is being monitored by a censoring agent controlled by the regime. The client wants to visit a censored (*covert*) destination, say cnn.com, by pretending that this communication is for an uncensored (*overt*) destination, say cat.com. The goal of the client is to unobservably communicate with the covert destination, i.e., the censoring agent is completely oblivious to the client's use of a circumvention tool to visit a censored destination. We assume that the censoring agent will check all incoming and outgoing traffic, and can easily block the name or addresses of suspicious circumvention proxies, such as ANDaNA [8] (a system similar to Tor) or VPN proxies.

Finally, we assume that the censoring agent knows about the existence and details of the circumvention tool. Also, the censoring agent knows the addresses of redirecting routers, but it cannot block the path including these routers (justified further in Section IV-D). This assumption is the opposite of existing proxy-based circumvention tools, which makes them easily blocked by their IP addresses in the current Internet or their domain names in ICN. We assume that the censoring agent does not actively manipulate the packets to compromise the client's privacy as that will cause significant collateral damage.

## III. Background and Related Work

### A. Background: Information-Centric Networks (ICN)

There are several information-centric network proposals of which Named-Data Networking (NDN) [9] has attracted much attention in recent times. Each ICN design is different, but all of them share a few basic traits such as routing based on names and the ability to leverage in-network caching as an optimization. There are two packet types in ICN, *interest* and *data*. An interest packet is a request for a specific name and each interest returns at most one data packet. For routing these packets, each router in the middle has three tables as follows:

- Content Store (CS): a table that stores cached data for future interests.

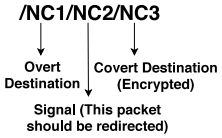


Fig. 1. Using different name components for router redirection in ICN

- Pending Interest Table (PIT): a table that stores interests awaiting matching content and their incoming interfaces.
- Forwarding Information Base (FIB): a table that stores outgoing interfaces to reach producers of specific names.

When an interest arrives, the router first looks up the CS for already cached data, and if the specific data is cached, the router drops the interest and sends the cached data on the incoming interface. If there is no match in the CS, it searches the PIT for pending interests and if there is a match, it means that another interest is waiting for the same content, so the router will add the incoming interface to that PIT entry. If there is no match, the router checks the FIB for a route to relay the interest and creates the corresponding PIT entry.

ICN is stateful for each packet at each on-path router, so that the matching data can traverse the reverse path of an interest. When the data arrives at a router, the router multicasts it along the incoming interfaces of all matching interests based on the PIT entry. The router then removes the PIT entry and caches the data in the CS for future queries.

### B. Related Work: Covert Communications in ICN

In order to use router redirection in ICN, users use covert communication to signal to routers enroute a communication path that the packets should be redirected. In a covert communication setting, two parties can communicate in an unobserved pattern without the third party noticing that a message is exchanged. Ambrosin et al. [10] list the covert communication channels in ICN. They list the channels in three categories:

- Delay-Based covert communication
- Common-prefix-based covert communication
- Errors and error Handling

The above work only focused on ephemeral covert channels, however, the main covert channel is the name components of interest packets. In ICN, a name consists of hierarchically structured components separated by "/". Several proposed systems [6], [11], [12] use random nonces or encrypted data in the name components for providing a service. These values can be used to exchange messages secretly without the censoring agent knowing that the client is using that specific service. Each name component can be seen as a cookie or a service nonce by the censoring router. We use different name components in our system for covert communication in ICN. Figure 1 shows an example of how we can use different name components as a covert communication channel.

Another covert communication channel in some ICNs such as NDN is the nonce field in each interest packet. For example in NDN, this nonce carries a randomly-generated 4-octet-long

byte-string. We can use this field for signalling the redirecting router that this interest packet should be redirected.

### C. Related Work: Decoy Routing Circumvention

Decoy routing [3]–[5] is a current Internet censorship circumvention approach motivated by the ease of IP address blocking of proxy-based circumvention tools. In a decoy routing protocol, a client will visit a blocked (covert) destination by showing the censoring agent that his request is for a non-blocked (overt) destination. This approach is based on some friendly ASes, called decoy ASes, that change their routers' routing tables in a way that by receiving tagged packets from users in a censored area, they will redirect them to the covert destination. These routers are called decoy routers. The decoy router protocols run as follows: (i) The client probes different overt destinations to find a path consisting of a decoy router; (ii) The client informs the decoy router by tagging the request packet that this packet needs to be redirected to a covert destination; (iii) The decoy router fetches the covert content and sends it back encrypted with a pre-shared key.

### D. Related Work: Censorship Circumvention Protocols in ICN

To the best of our knowledge, this work is the first to use traffic redirection for evading censorship in ICN. The main difference between our approach and previous works on censorship circumvention in ICN is the *unobserved* traffic of the client. In unobserved communication, the censoring agent cannot detect that the client is using our approach to get a blocked content or it is communicating with an uncensored destination. We can categorize existing ICN privacy proposals into three main categories: Table I shows a summary of their ideas and drawbacks, as elaborated next.

**Using no proxy:** In these systems, there is no proxy in the middle of the connection between the consumer and the producer. Arianfar et al. [13] proposed a new scheme for obfuscating content names in ICN with no proxy in the middle of the connection. In this approach, the content provider chooses a random cover file the same size as the real content file and XOR the two files after splitting them into chunks. The content provider then publishes the encoded chunks into the network. The names of these encoded chunks are a mix of hashed names and hash of cover chunks. The content provider will send the metadata consisting of the content hash, content length, the cover file, the names, and the name generation algorithm in a secure channel. The client can request these names and can decipher them upon retrieval. The main problem with this approach is that the communication overhead is 100% i.e. for retrieving a file, the client must receive a cover file of the same size in a secure channel.

Elabidi et al. [14] proposed a privacy-preserving extension to ICN by providing a mechanism to stop dissemination after identity expiration. In addition to network elements, we have three more entities in this scheme: (i) Identity providers that issue expiring identities for the network entities so they can communicate. (ii) Trust verification providers will be asked to verify one identity and its expiration date. (iii) Digital identity protection authorities which be informed if one entity uses an expired identity for communication. Fotiou et al. [1] proposed a privacy scheme for ICN by using homomorphic encryption in a hierarchical brokering system. The producers submit their contents in this system organized as a tree, and the consumers send an encrypted query to the root of the tree. This query is answered with a pointer to the producer with just homomorphic operations without any decryption.

**Using one proxy:** In these systems, the consumer sends the interest to a proxy in the middle, and the proxy sends a new interest to the producer. These protocols use coding techniques, e.g., Tao et al. [15] use random linear network coding (RLNC). In this protocol, the consumer and the producer split the interest and the data into multiple chunks and send their linear combination. Another scheme in this category is proposed by Tourani et al. [2] that use Huffman coding. In this scheme, each consumer shares a Huffman coding table with a proxy (*anonymizer*), and the consumer encodes the interests with this table and sends them to the network.

**Using two proxies:** For providing anonymity, some schemes use onion routing similar to Tor but just with two proxies in the middle of connection. ANDaNA [8] is a censorship circumvention protocol using two proxies wherein one sees the requester identity and the other sees the content name, so without colluding, they cannot relate the content name to its requester identity. Chung et al. [16] proposed a similar approach to ANDaNA using two proxies wherein the user encrypts the interest with two symmetric keys in an onion manner. The main difference of this protocol with ANDaNA is that a hash of the name is embedded in the first layer of the onion to enable cache utilization.

## IV. Router Redirection in ICN

In the following, we provide an overview of how our protocol for censorship circumvention works. Figure 2 shows the scenario that a client uses redirecting router. Our protocol based on router redirection has two phases:

1) **Registration:** The censored client should register in the router redirection system to inform the redirecting router about its interest packets.
2) **Traffic Redirection:** Now that the redirecting routers are aware of the tags inside the interest packets of registered users, they can redirect them to another destination.

### A. Registration in Router Redirection

In this phase, the client submit its credentials used to generate the interest packets to covert destinations, which prompts the registering server to update the configuration of the redirecting routers for the new client. The goal of this phase is that the redirecting routers only deflect the packets of the users registered in the system, but will leave other packets unchanged. The client can inform the system of its credentials thorough a latency-insensitive communication channel such as email or social networks. These credentials are encrypted with the public key of the registration server.

TABLE I
RELATED WORK ON CENSORSHIP CIRCUMVENTION IN ICN

| Paper | Number of proxies | Idea | Drawback |
|---|---|---|---|
| Arianfar et al. [13] | No proxy | XOR the content with a random cover file and using hash of names | 100% communication overhead over the secure channel |
| Elabidi et al. [14] | No proxy | Providing expireable identities for users and existence of authority entities | Adding three more entities to the network and requiring more rounds of interactions |
| Fotiou et al. [1] | No proxy | Homomorphic encryption for retrieving names in an hierarchical brokering system | High computation time for homomorphic operations |
| Tao et al. [15] | One proxy | Encoding the interest packet with random linear network coding (RLNC) | Processing time of using asymmetric operations and RLNC |
| Tourani et al. [2] | One proxy | Encoding the interest packet with Huffman coding | Censoring agent can block the anonymizer's domain name (plain text) |
| Dibenedetto et al. [8] | Two proxies | Onion routing similar to Tor | High delay time, and the first proxy can be blocked by the censoring agent |
| Chung et al. [16] | Two proxies | Onion routing similar to Tor and embedding a hash of name in the first layer for providing cacheability | High delay time, and the first proxy can be blocked by the censoring agent |
| Our approach | No proxy | Router redirection using ephemeral names or one-on-one protocols such as key exchange protocols | |

In the following sections, we introduce two different approaches for interest traffic redirection in ICN.

### B. Traffic Redirection: Using Ephemeral Names

A fundamental problem with static names in ICN is that the censoring agent can create a blacklist of blocked destinations, and filter all the packets that are going out or coming inside the censored region. If the domain or first name component matches an entry in the blacklist, the censoring agent drops that packet. Therefore, if the clients use ephemeral names that change periodically, the censoring agent cannot make such a blacklist of censored domains. For traffic redirection using ephemeral names, the redirecting router has more than one name. Each time a new client registers in the system, a new name will be created for the redirecting router. These ephemeral names will be generated as follows:

$$EN = HMAC(N_R \mid k_R) \tag{1}$$

where $N_R$ is the global name of the redirecting router, $\mid$ is concatenation, $k_R$ is the client's private key submitted in registration phase. The redirecting router's FIB table is updated and for each of these names, a route is entered. If one of the redirecting routers receives an interest where the first name component is the *EN* of a redirecting router, they redirect the interest to the covert destination by replacing the first name component with the actual name of the covert destination. The second name component is the covert destination name encrypted with the secret key of the client.

This approach addresses privacy concerns in ICN as follows:

- **Ephemeral names:** The names are generated periodically, and the censoring agent can not detect tagged packets if it does not know the private key of the user.
- **Ephemeral encrypted content:** When the redirecting router receives an interest, it returns the blocked data from the covert destination encrypted with $k_R$.
- **Ephemeral signature:** The redirecting router generates a public-private key pair for each consumer based on $k_R$ that enables the specific client to derive the same keys to verify the authenticity of the packets.

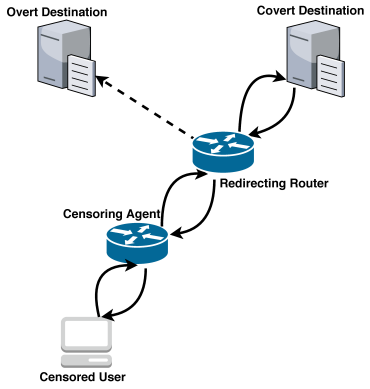


Fig. 2. Routing redirection in ICN using ephemeral names

**Cache Utilization:** For enabling caching at on-path routers from the client to the redirecting router, we can change two parameters in this system for future interests:

- The period after which ephemeral names (*EN*) are replaced with a new name. If the ephemeral names are generated for long periods, the name of redirecting router will be fixed for that period, and the on-path routers will cache the blocked content for future use without knowing they are caching censored data.
- The number of users that can use a shared consumer secret key or $k_R$. If more than one client uses a private key, the ephemeral name of the redirecting router (*EN*) is the same for them. By assigning a private key to a group of users, the overhead at the redirecting router decreases too, and users in one group can leverage caching at on-path routers.

### C. Traffic Redirection: Using a Key Exchange Protocol

Unlike integrity and authenticity, confidentiality is ignored in ICN and is treated as an application layer feature. Mosko

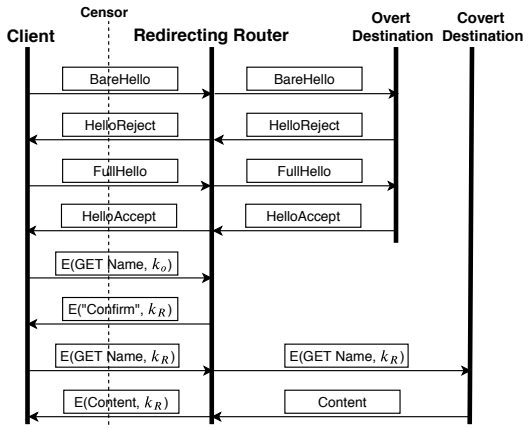


Fig. 3. Routing redirection using a key exchange protocol. $k_o$ is the shared key between overt destination and client, and $k_R$ is the pre-shared key between client and redirecting router.

et al. [6] proposed the first ICN key exchange protocol to enable encrypted sessions between consumers and producers. This scheme needs at least two RTT for creating a secure session between the producer and the consumer, and it adds 30% more delay to the connection. The redirecting router can be informed with the nonce embedded in this protocol that this session should be hijacked. Thereafter, the consumer send the covert destination name via this secure session to the redirecting router.

**Tag generation.** After a client registers in the system, the redirecting router generates the tags that this client will use for its packets. The tag inside the interest packet shows that the packet belongs to a censored user, so its packets should be redirected. These tags are generated in real-time as follows:

$$tag = HMAC(k_R \mid time) \tag{2}$$

where $\mid$ means concatenation, and $k_R$ is a private key that the censored client submitted in the registration phase. This tag changes based on time, i.e., it acts as pseudo-random number generator.

As shown in Figure 3, the client uses CCNxKE [6] to communicate with the redirecting router to fetch blocked content via the following steps:

1) **BareHello:** The client sends a Barehello message with the destination of overt destination that is not blocked. This message obtains a source challenge that is a random number to bind the session to this client. This challenge is needed since there is no source address in ICN, so CCNxKE protocol uses this challenge as a proof of the origin of the session. The client puts a tag (generated by Equation 2) inside this field to inform the redirecting router that this packet needs redirection.
2) **HelloReject:** The overt destination returns public information about itself.
3) **FullHello:** The client starts a key exchange protocol by sending its key share. The client also sends a source proof to show that it is the same entity that started the session.
4) **HelloAccept:** The overt destination sends back its key share in addition to a session ID. After this round, both parties—the overt destination and the censored client—can construct a shared key, $k_o$, with Diffie Hellman pairs they exchanged.
5) **Check for redirecting router:** The client sends an interest for a content in overt destination.
6) **Redirecting router presence:** The redirecting router sends back a confirmation response encrypted with a pre-shared key ($k_R$) that is submitted by the client in registration phase.
7) **Request for blocked content:** The client asks for blocked content in the covert destination encrypted with the pre-shared key, $k_R$. The redirecting router will decrypt the interest name and send a new interest for the blocked content.
8) **Blocked content:** The redirecting router will fetch the blocked content, encrypt it with $k_R$, and send it back to the client.

After four rounds, the client gets the blocked content in the covert destination using CCNxKE protocol. The censoring agent in the middle cannot detect that the client is communicating with a censored destination since the censoring agent does not have $k_R$ used in generating tags. Therefore, the censoring agent cannot distinguish the random challenge in the BareHello from a *tag* used for censorship circumvention.

### D. Routing Around Decoy (RAD) Attack

Existing decoy routing protocols in the current Internet are vulnerable to specific routing attacks by the censoring agent, called routing around decoys (RAD) [17]. In this attack, the censoring agent will tamper the BGP routes, so the traffic of the censored users will not pass through the decoy routers.

However, an advantage with ICNs like NDN is that the censoring agent cannot block a path including a redirecting router since usually each router has an interface for each address in its FIB that only includes one step further. Therefore, each router can see one hop after itself, and it does not have the power to block a path contains a redirecting router.

## V. Experiments

**Experiment Configuration** We have simulated our protocol in ndnSIM [7], an NS-3 based Named Data Networking (NDN) simulator. In ndnSIM, all the packet are in NDN format, and all the forwarding and management strategies are implemented directly using the source code of Named Data Networking Forwarding Daemon (NFD). For the links in the network, we choose 10Mbps and 1ms as bandwidth and delay. We simulate the scenario for 100 clients that each request for a file in each second, and the runtime is 10 seconds for the simulations. We only simulate traffic redirection using ephemeral names (Section IV-B). For this experiment, we measure the time of requesting a file and transferring from

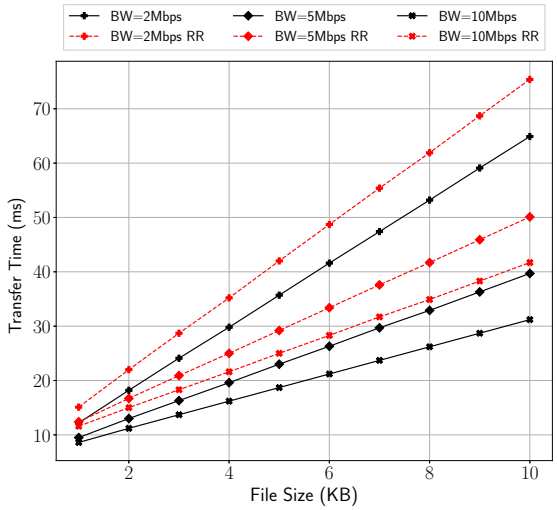


Fig. 4. Transfer time for NDN with and without the router redirection (RR) approach and for different file sizes and client bandwidths

the server, so we assume that the client has already registered for the router redirection system. Our metric for comparing the performance of our protocol is content download time.

**Results** Figure 4 shows the result for downloading a file with different sizes and with different client bandwidths. This figure also shows the file transfer time for scenarios with and without our protocol. The figure shows that when the client uses router redirection, the transfer time increases slightly. For instance, for a file of 10MB, the transfer times will be 31.2ms and 41.7ms without and with router redirection, respectively, when the client uses 10Mbps bandwidth. Therefore, we see that **for a modest delay overhead (15%-50%), the client can retrieve the file evading the censoring agent**.

## VI. Conclusion and Future Work

We presented router redirection, a censorship circumvention technique in information-centric networks (ICN). In this approach, a client located at a censored region notifies a friendly router in the path of communication that its packets need to be redirected to another destination. This communication is feasible since the client uses the covert communication fields inside the interest packet. We describe our design and discuss different options for the client to evade the censoring agent. We also evaluate our design by experimenting in ndnSIM [7], a well-known simulator based on NS3 for named data networking (NDN). The metric in our experiments is file transfer time, and we have evaluated our design for different file sizes and bottleneck bandwidths.

As part of future work, we plan to use more sophisticated techniques for the registration phase. Furthermore, we plan to make downstream traffic unobservable. After fetching the covert destination's content, the redirecting routers should send back the data to the requesting client in a way such that the censoring agent cannot detect the difference between the overt and covert contents. We believe that router redirection is a major step towards making blocking-resilient communications in ICN.

## References


[1] Nikos Fotiou, Dirk Trossen, Giannis F Marias, Alexandros Kostopoulos, and George C Polyzos. Enhancing information lookup privacy through homomorphic encryption. *Security and Communication Networks*, 7(12):2804–2814, 2014.

[2] Reza Tourani, Satyajayant Misra, Joerg Kliewer, Scott Ortegel, and Travis Mick. Catch me if you can: A practical framework to evade censorship in information-centric networks. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, pages 167–176. ACM, 2015.

[3] Amir Houmansadr, Giang TK Nguyen, Matthew Caesar, and Nikita Borisov. Cirripede: Circumvention infrastructure using router redirection with plausible deniability. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 187–200. ACM, 2011.

[4] Josh Karlin, Daniel Ellard, Alden W Jackson, Christine E Jones, Greg Lauer, David Mankins, and W Timothy Strayer. Decoy routing: Toward unblockable internet communication. In *FOCI*, 2011.

[5] Eric Wustrow, Scott Wolchok, Ian Goldberg, and J Alex Halderman. Telex: Anticensorship in the network infrastructure. In *USENIX Security Symposium*, 2011.

[6] Marc Mosko, Ersin Uzun, and Christopher A Wood. Mobile sessions in content-centric networks. In *IFIP Networking Conference (IFIP Networking) and Workshops, 2017*, pages 1–9. IEEE, 2017.

[7] Alexander Afanasyev, Ilya Moiseenko, Lixia Zhang, et al. ndnsim: Ndn simulator for ns-3. *University of California, Los Angeles, Tech. Rep*, 4, 2012.

[8] Steven DiBenedetto, Paolo Gasti, Gene Tsudik, and Ersin Uzun. Andana: Anonymous named data networking application. *arXiv preprint arXiv:1112.2205*, 2011.

[9] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12. ACM, 2009.

[10] Moreno Ambrosin, Mauro Conti, Paolo Gasti, and Gene Tsudik. Covert ephemeral communication in named data networking. In *Proceedings of the 9th ACM symposium on Information, computer and communications security*, pages 15–26. ACM, 2014.

[11] Cesar Ghali, Marc A Schlosberg, Gene Tsudik, and Christopher A Wood. Interest-based access control for content centric networks. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, pages 147–156. ACM, 2015.

[12] Eslam G AbdAllah, Mohammad Zulkernine, and Hossam S Hassanein. Preventing unauthorized access in information centric networking. *Security and Privacy*, 1(4):e33, 2018.

[13] Somaya Arianfar, Teemu Koponen, Barath Raghavan, and Scott Shenker. On preserving privacy in content-oriented networks. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pages 19–24. ACM, 2011.

[14] Amine Elabidi, Ghazi Ben Ayed, Sonia Mettali Gammar, and Farouk Kamoun. Towards hiding federated digital identity: Stop-dissemination mechanism in content-centric networking. In *Proceedings of the 4th international conference on Security of information and networks*, pages 239–242. ACM, 2011.

[15] FENG Tao, XING Fei, Lu Ye, and Fang Jun Li. Secure network coding-based named data network mutual anonymity communication protocol. In *Proceedings of International Conference on Electrical, Computer Engineering and Electronics (ICECEE)*, pages 1107–1114. Citeseer, 2015.

[16] Seog Chung Seo, Taehong Kim, and Myeongwuk Jang. A privacy-preserving approach in content centric. In *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*, pages 866–871. IEEE, 2014.

[17] Max Schuchard, John Geddes, Christopher Thompson, and Nicholas Hopper. Routing around decoys. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 85–96. ACM, 2012.