

Defend against Internet Censorship in Named Data Networking

Xingmin Cui*, Yu Hin Tsang*, Lucas C.K.Hui*, S.M.Yiu*, Bo Luo†

*The University of Hong Kong, Hong Kong

{xmcui, yhtsang, hui, smyiu}@cs.hku.hk

†Beijing University of Posts and Telecommunications, China

13126645140@163.com

Abstract—Named Data Networking (NDN), as a promising next generation network, has several advantages in scalability, mobility and network efficiency compared with IP-based Internet. These advantages come with the usage of named content, powerful routers and intermediate cache. These, however, make censorship in NDN more easily to deploy. In this paper we give an analysis of the possible censorship techniques in NDN and come up with two most effective attacks: name-watchlist attack and deep packet inspection. Existing solutions against censorship in IP-based network and NDN either cannot defend against both attacks or disable the cache mechanism, thus decrease the network efficiency. We propose a scheme which adopts the idea of proxy web servers and brings in the concept of smart routers to partially persevere the functionality of intermediate cache and improve network performance. Security analysis shows that as long as the consumer can connect to a smart router which performs a second-time disguise of the requested name without going through a compromised smart router, he can retrieve his interested Data packet. Simulation result shows that the usage of smart routers helps mitigate the performance overhead to disguise Interest names and Data content.

Index Terms—Named Data Networking, Internet censorship, anti-censorship, proxy server, smart router

I. INTRODUCTION

Named data networking (NDN) [16] is one promising next generation network candidate to deal with the scalability, mobility and security issues of today's Internet. NDN uses names instead of IP addresses to identify packets in the network, including Interest packets and Data packets. NDN packets are valid no matter where they come from or where they are forwarded to. Therefore NDN uses content store(CS) to cache received Data packets to satisfy future Interests requesting the same name. This increases network efficiency since Interest packets do not need to be forwarded to the producer. However, as pointed out in [6], the named content and intermediate routers with bigger computational and memory resources makes Internet censorship more easily to deploy in NDN.

Internet censorship and anti-censorship techniques have been well studied in IP-based network[10], [11], [15]. But these solutions are not readily applied to NDN because of the different design paradigm of NDN and IP. For example, some anti-censorship solutions in IP-based network [10] propose to let the requesters and responders communicate over a covert channel. But NDN lays emphasis on the distribution of content rather than host-to-host communication. Existing

anti-censorship techniques proposed in NDN either disables the functionality of content store [2] or reduces NDN to IP-based network [8].

In this paper we aim to construct an anti-censorship system which defends against Internet censorship and at the same time preserves the advantages of NDN (eg. content store). We consider the scenario of state-level censorship [15] where data producers and certain part of the network are not under control of the censor. We first analyze the feasibility of existing Internet censorship techniques in NDN and find out that among others, two censorship techniques are most effective: *name-watchlist attack* and *deep packet inspection (DPI)*. In name-watchlist attack, the censor has a list of censored names. He inspects the names of each packet passing by and once he finds out that the name is a censored name or the transformation of a censored name, he will suppress or delete this packet. In deep packet inspection, the censor installs DPI devices to compromised nodes to inspect the packets passing by and check whether they contain sensitive data [11].

Our scheme disguises the censored name into valid names to defend against name-watchlist attack and encrypts the replied data packets to defend against deep packet inspection. It adopts the idea of proxy web server and brings in smart routers to partially preserve the functionality of intermediate routers' content store. The contributions of this paper are as follows:

- We propose an anti-censorship scheme which defends against the two most effective censorship techniques in NDN while partially preserves the functionality of intermediate content store.
- We give security analysis of the system indicating the condition on which consumers succeed in getting their requested data. Simulation shows that smart routers enables the usage of intermediate content store and mitigates the overhead to disguise Interest names and Data content.

The rest of the paper is organized as follows: In section II, we will discuss the feasibility of existing censorship techniques in NDN. We give our system model and security model in section III. In section IV, we introduce our anti-censorship system. Security analysis and performance evaluation will be given in section V. Related work is listed in section VI and section VII concludes the paper.

II. INTERNET CENSORSHIP

Internet censorship is used to control and suppress the access to, or the publishing of certain information on the Internet [11]. It is often used to prevent censored users from accessing sensitive content, which resides in the uncensored Internet. Censorship and anti-censorship techniques have been well-studied in IP-based network [11], [10], [5], [15]. Censorship policies need to enforce censorship without causing performance degradation of the network. Software-based censorship technologies include IP blocking, DNS tampering, keyword filtering, monitoring the usage of specific protocols, etc. Hardware-based measures include using Deep Packet Inspection(DPI) devices to inspect the packets passing by [11]. The objective of an anti-censorship system is to connect censored users to the uncensored Internet securely and anonymously [11].

Since NDN packets are identified by names instead of IP addresses, the counterpart of *IP blocking* in NDN would be *name-watchlist attack*[2]. In a name-watchlist attack, the censor has a list L , which contains the names he wishes to block or eliminate. We call these names *censored names*. The sensor inspects the names of each packet passing by and once he finds out that the name is a censored name or transformation of a censored name, he will suppress or delete this packet.

The counterpart of *DNS tampering* in NDN would be *name tampering*, that is, to modify the name of a packet. However name tampering would be much more difficult to employ due to the following reasons: Firstly, there doesn't exist a central name resolution server in NDN. Secondly, data consumers can easily check whether the received Data packet is in reply to the name he has requested since NDN requires the Data packet to enclose producer's signature on the packet name and content.

Deep packet inspection and *keyword filtering* are still applicable in NDN. In fact, intermediate cache and routers with bigger computational and memory resources make them more easily to deploy. Keyword filtering can be achieved through deep packet inspection.

From the above discussion we can see that in NDN, the most promising censorship techniques are **name-watchlist attack** and **deep packet inspection(DPI)**. This paper aims to come up with an anti-censorship scheme to defend against these two attacks. Here notice that name-watchlist attack violates *name privacy* and DPI violates *content privacy*. Simply encrypting data packets only protects content privacy but cannot protect name privacy.

III. SYSTEM MODEL AND SECURITY MODEL

A. System model

We follow the model of [15], which they call state-level censorship and adapt it to NDN. In this model, producers and consumers reside in different networks. To save efforts, the censor usually controls the backbone routers that link the consumers and producers together.

Three parities exist in the network: data producers, consumers and routers. Consumers send out Interest packets to

request for Data packets. Intermediate routers forward Interests towards the producer. Producers, after receiving the Interest, reply it with the corresponding Data packet. Then the Data packet is routed back to the consumer by intermediate routers, taking the reverse way of the Interest packet. Among all the routers, we assume the existence of a certain amount of smart routers (SR). The motivation to bring in smart routers and their capabilities will be given in section IV.

B. Security Model

The censor can carry out the following actions:

- *Control a subset of network entities*: The censor can control a subset of routers and consumers in his network and perform any actions allowed to these entities. Censored entities can communicate with each other and exchange information to strengthen the censorship.
- *Compromise existing routers*: The censor can select any router in his network and compromise it. Then the censor learns all the private information (eg. private key, cache content) of this router. But we assume the censor does not have the power to control all routers in the network.
- *Compromise existing consumers*: In this case, the censor can act as a normal consumer to request interested data.

We assume compromised routers can launch name-watchlist attack and DPI to prevent consumers from retrieving sensitive information from producers. Besides, we assume the existence of a Public Key Infrastructure (PKI) in the system. This assumption is reasonable since NDN requires producers to sign on the $\{\text{name}, \text{data}\}$ tuple to ensure the integrity of Data packets which implies the existence of a PKI. Besides, censorship should NOT violate the availability of legitimate data [13]. For example, a censor cannot simply delete all the packets since he may delete legitimate packets as well.

IV. A CENSORSHIP RESISTANT SYSTEM

A. Design Goals

Our proposed censorship resistant system aims to achieve the following goals:

- *Against name-watchlist attack*: Censored names need to be disguised into random-look patterns to escape this attack.
- *Against DPI*: It should be difficult for a censor to claim that a Data packet contains censored content.
- *Performance Requirement*: The system should not disable the cache mechanism which decreases the network efficiency.

Other requirements such as tamper resilience and anonymity are not our focus since solutions already exist for them. The signature on the $\{\text{name}, \text{data}\}$ tuple ensures temper resilience. ANDaNA [14] can be used to ensure anonymity.

B. Cache, Smart Router and Proxy

Censored names need to be disguised to escape censorship. In this paper we denote the disguised names as *valid names* and the censored name as the *original name*. As mentioned before, routers in NDN cache Data packets in their content

store to satisfy future requests for the same name. Therefore intermediate routers should be able to discover that two valid names correspond to the same original name to make use of the content store. If this router happens to be compromised, the censor can easily tell whether the current requested name is actually a censored name. Besides, a compromised router can deploy DPI to the Data packets in its cache and mark those with sensitive keywords as censored. When a later Interest asks for sensitive Data packets, it will block the request. For the above reasons we conclude that fully making use of cache will decrease the effectiveness of anti-censorship.

To achieve the goal of anti-censorship and at the same time preserve the high network efficiency, we assume the existence of *smart routers* to facilitate the network flow. Producers communicate with smart routers in advance to negotiate some secret information. After that the smart router can use this information to recover the original name from valid names. In this way, it can tell whether two disguised names correspond to the same original name. A smart router can reside in the censor's network (which we call *inside SRs*) or outside of the censor's network (which we call *outside SRs*). Outside SRs act like proxies in today's network. They can retrieve Data packets using the original name without being blocked since they are not under the censor's control.

We claim that our assumption of the existence of smart routers is reasonable since this concept is similar to the concept of proxy in IP-based network, which has been adopted by related research [7], [11].

C. System Design

1) *System Setup*: In the setup phase, each smart router communicates with the producer to get the proxy re-encryption key [3]. The re-encryption key $RK_{Prod \rightarrow SR}$ is generated by the producer given SK_{Prod} and PK_{SR} and sent to the smart router.

2) *Data request process*: **Consumer disguises the censored name and sends out Interest packet**: To request the Data packet with a censored name N_c , the consumer first picks a random value $salt$, concatenates it with N_c and encrypts it using the producer's public key. The resulted name is in the format $N_v = Enc(PK_{Prod}, N_c || salt)$. After that the consumer constructs an Interest packet with name N_v and sends it out. This $salt$ value would be used by the producer to encrypt the replied Data packet.

Intermediate Routers route the Interest: As mentioned before, we assume the existence of smart routers. We denote the remaining routers as *normal routers*. A normal router does not have the capability to recover the original name of N_v , and there is a negligible probability that the cached Data packets have the same name with the incoming Interest. Therefore it can only flood this Interest to all the outgoing faces.

For a smart router, however, since it is possessed of the re-encryption key $RK_{Prod \rightarrow SR}$, he can do re-encryption and decryption to recover the original name. Besides, like in IP-based network where users in the censored network are aware

of a list of proxies in the uncensored network, we assume that inside SRs are aware of a list of outside SRs (or proxies).

We assume the censor has the power to control backbone routers, which in the worst case can act as smart routers and request for re-encryption keys from the producer. In this case, the Interest with name $Enc(PK_{Prod}, N_c || salt)$ would be blocked when it arrives at the censored backbone. To avoid this, the inside SR randomly selects one outside SR, further encrypts N_v using this outside SR's public key. It can also choose not to do this and directly forward the Interest with name N_v to the producer. The actions of an inside SR are listed in Algorithm 1 and the actions of an outside SR (proxy) are listed in Algorithm 2.

Algorithm 1 Actions of an inside SR to route Interest

Require: On receive Interest packet with name N_v

- 1: $N_v = C_{Prod} = Enc(PK_{Prod}, N_c || salt)$
 - 2: $C_{SR} = Re-enc(N_v, RK_{Prod \rightarrow SR})$
 - 3: $N_c || salt = Dec(SK_{SR}, C_{SR})$
 - 4: **if** Data of N_c in cache **then**
 - 5: Name = N_v
 - 6: Content = $Enc(SK_{SR}, N_v || Enc(SK_{Prod}, N_c || data))$
 (SR's signature on name||data)
 - 7: Payload = $Enc(salt, Content)$
 - 8: construct a new Data packet with (Name, Payload) and return to consumer
 - 9: **else** {Data of N_c not in cache}
 - 10: add the *incoming face* to the PIT entry of N_c (create one if not exist yet)
 - 11: randomly choose a bit b from $\{0, 1\}$
 - 12: **if** $b = 0$ **then**
 - 13: randomly select an outside SR Pxy , compute $N'_v = Enc(PK_{Pxy}, N_c || salt)$
 - 14: record the mapping $\langle N'_v, N_v, N_c, salt \rangle$
 - 15: construct an Interest with name N'_v and forward it towards Pxy
 - 16: **else** $\{b = 1\}$
 - 17: record the mapping $\langle N_v, N_c, salt \rangle$
 - 18: forward the Interest with name N_v towards the producer
 - 19: **end if**
 - 20: **end if**
-

Producer Replies with Data packet: If the Interest packet cannot be satisfied by intermediate routers, it will arrive at the producer. In this case, the Interest must have been forwarded from an outside SR with name N_c . The producer will reply with a Data packet with name N_c and payload $Enc(SK_{Prod}, N_c || data)$ (producer's signature on name||data).

Intermediate Smart Routers cache and forward the Data packet: When the replied Data packet with name N_c arrives at an outside SR, it traverses its record to find the mapping $\langle N'_v, N_c, salt \rangle$ (it always exists since Data packets take the reverse way of Interest packets). Then it stores the tuple $(N_c, Enc(SK_{Prod}, N_c || data))$ in its content store, constructs

Algorithm 2 Actions of an outside SR (proxy) to route Interest

Require: On receive Interest packet with name $N'_v = Enc(PK_{Pxy}, N_c || salt)$
 $N_c || salt = Dec(SK_{Pxy}, N'_v)$

- 2: **if** Data of N_c in cache **then**
 Name = N'_v
- 4: Content = $Enc(SK_{Pxy}, N'_v || Enc(SK_{Prod}, N_c || data))$
 (SR's signature on name || data)
 Payload = $Enc(salt, Content)$
- 6: construct a new Data packet with (Name, Payload) and return to the inside SR
- else** {Data of N_c not in cache}
- 8: add the *incoming face* to the PIT entry of N_c (create one if not exist)
 record the mapping $\langle N'_v, N_c, salt \rangle$
- 10: construct an Interest with name N_c and send it towards the producer
- end if**

a Data packet following lines 3-5 in Algorithm 2 and sends it to back to the *incoming faces* of the PIT entry with name N_c (which has been stored by line 8 in Algorithm 2).

When the replied Data packet of name N'_v arrives at the inside SR, it traverses its record to find the mapping $\langle N'_v, N_v, N_c, salt \rangle$. Then it decrypts the received Data packet using $salt$ and recovers the payload. It verifies the signature P_{xy} and stores the tuple $(N_c, Enc(SK_{Prod}, N_c || data))$ in its content store. Finally it constructs a new Data packet following lines 5-7 in Algorithm 1 and sends it back to the consumer (to the *incoming faces* of the PIT entry of N_c which has been store on line 10 in Algorithm 1)

Consumer verifies the replied Data packet: Following our design, the received Data packet must be forwarded back to the consumer from an intermediate smart router. Name of the Data packet is N_v and payload is $Payload = Enc(salt, Content)$, where $Content = Enc(SK_{SR}, N_v || Enc(SK_{Prod}, N_c || data))$. The consumer first recovers $Content$ by decrypting $Payload$ using $salt$. After that he needs to do two levels of verification: The first level is to verify if the outer part of $Content$ is a valid signature of SR and N_v is his requested name. If succeeds, he continues to verify if the inner part $Enc(SK_{Prod}, N_c || data)$ is a valid signature of the producer and N_c is the original name of N_v . He only accepts the Data packet when both levels of verification succeed.

V. ANALYSIS

A. Security Analysis

Suppose a consumer sends out an Interest with name N_v , which is the disguise of a censored name N_c . The forwarding strategy of NDN allows an Interest packet to be forwarded to multiple outgoing faces, therefore several paths exists from the consumer to producer. In our scheme, smart routers can choose to further disguise the received name N_v to N'_v or simply forward the Interest with name N_v . This second-time disguise

TABLE I
TEST CASES

Case	Scenario	Case	Scenario
1	No smart routers	6	SR: 1 router in group2
2	SR:1 router in group1	7	SR: 2 routers in group2
3	SR:2 routers in group1	8	SR: 4 routers in group2
4	SR:3 routers in group1	9	SR: all routers(6) in group2
5	SR:all routers(4) in group1	10	No Censorship

can only happen once because other SRs cannot recover the original name after the disguise.

We denote the smart router which performs the second-time disguise as *Se-enc-SR* (since this SR does a *Second time encryption* of N_v). If no compromised smart router exists between the consumer and *Se-enc-SR*, the consumer can get his requested data. This is because once *Se-enc-SR* further disguises N_v to N'_v , other routers including smart routers cannot recover N_c from N'_v . Thus it has no reason to block N'_v . The probability at which consumers succeed in getting their requested data relies on where to locate the smart routers and which SR acts as the *Se-enc-SR*. We defer the exploration of how to increase this probability as our future work.

B. Performance Evaluation

In our scheme, we use smart routers to partially preserve the functionality of content store. Smart routers from the consumer to *Re-enc-SR* and routers from the proxy to the producer can make use of their content store. We constructed a system with state-level censorship using the NDN simulation platform ndnSIM [1] and applied our anti-censorship scheme on it.

To simulate state-level censorship, we designed a topology which consists of 5 producers, 20 consumers and 22 intermediate routers. Consumers are in the censor's network and producers are out of it. 6 of the 16 inside routers are compromised, among which 2 are backbone routers. Smart routers use the scheme proposed by Ateniese et al.[3] to perform proxy re-encryption.

The simulation aims to study the performance improvement bring by the usage of content store. We divide compromised routers into three groups: routers close to the backbone (group 1), routers close to consumers (group 2) and others (group 3). To find out the influence of number and location of smart routers, we tested ten scenarios. Details of these scenarios are listed in table I. We performed four rounds of simulation, the number of Interests per consumer (IPC) varies in each round from 10 to 40. The simulation results are given in figure 1.

From figure 1 we have these observations: (1) With the increase of SRs, the average file retrieval time decreases. (2) When SRs locate closer to the consumer, the effectiveness of content store is more obvious. In summary, the result shows that although our anti-censorship scheme requires extra overhead to disguise censored names and encrypt replied data, the idea of smart routers enables the usage of intermediate cache and mitigates this extra overhead.

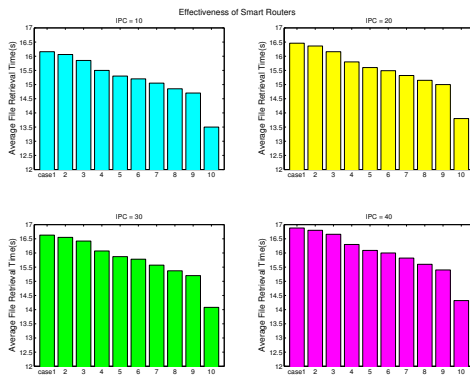


Fig. 1. Simulation Results

VI. RELATED WORK

As one of the most promising next generation network architecture, the security and privacy issues of NDN has attracted lots of researchers' attention [6], [12], [14], [2]. [6] gives an overview of the privacy issues in Content-Oriented Networking (CON), which has the same design paradigm with NDN. It points out Internet censorship appears to be easier in CON architectures because of the naming content, routers with bigger computational and memory resources and unencrypted interests and data. A wide range of anti-censorship techniques have been developed in IP-based network and NDN including user anonymization, covert channel, deniable publishing, proxy web server, etc [11]. Next we briefly summarize these solutions.

Tor [9] is a widely used tool to protect user anonymity. It uses onion routing and layered encryption to build a multi-hop circuit and forward users' requests through the circuit. ANDaNA [14] is the equivalent of Tor in NDN. But in NDN content is central to the network rather than hosts. Therefore content is the target of censors. Only protecting user anonymity is not enough to defend against censorship in NDN.

Infranet [10] uses covert channel to defend against censorship in IP-based network. Requesters and responders communicate over the covert channel. But this end-to-end communication paradigm is not in accordance with the design principles of NDN.

[6] suggests to use bloom filters to hide the keywords in names to achieve name privacy. But the bloom filter solution has several disadvantages. First, the adversary is also aware of how to use a bloom filter to represent a name. Therefore he can block the bloom filters representing the censored names. Second, the producer, intermediate routers and consumers need to agree on the hash functions to compute the bloom filter. Using the same set of hash functions for all names in the system would result in a large false positive rate [4].

[2] utilizes steganography to defend against name-watchlist attack and content analysis. It leverages computational asymmetry by forcing the adversary to perform sizeable computations to reconstruct the user's request. But in their scheme, the

data user needs to ask for several blocks in order to get one block of the target file which brings a lot of extra network overhead. Besides, the content store of intermediate routers cannot be used since the success of their scheme to defend against censorship relies on the fact that different consumers will construct different mixed blocks and cover blocks to recover the target blocks.

Our previous work [8] studies the effectiveness of censorship in NDN and discovers that although NDN benefits from the *multiple outgoing faces* forwarding strategy to disseminate information, anti-censorship is still needed especially when there are multiple censors in the network. [8] also proposes to disguise censored names into valid names, but it relies on an extra prefix to do routing. In this way it reduces NDN routing to a host-centric routing as in IP. The censor can block all the names with a given prefix, just like IP blocking in IP-based network. The proposed scheme in this paper does not have this problem.

VII. CONCLUSION

In this paper we propose a scheme to defend against Internet censorship in NDN. We focus on two types of attacks: name-watchlist attack and deep packet inspection, which, are NDN's most effective censorship techniques.

Our scheme adopts the idea of proxy web servers and brings in smart routers to make use of intermediate cache and increase network efficiency. Security analysis shows that as long as the consumer can connect to a smart router which performs a second-time disguise of the requested name without going through a compromised smart router, he can retrieve his interested Data packet. Simulation result shows that the usage of smart routers indeed helps mitigate the performance overhead to disguise Interest names and Data content.

ACKNOWLEDGEMENT

The work described in this paper was partially supported by the HKU Seed Fundings for Applied Research 201409160030; HKU Seed Fundings for Basic Research 201311159149 and 201411159122; National Natural Science Foundation of China (No.61572157), PRC; and Shenzhen Strategic Emerging Industry Development Foundation (No.JCYJ20150403161923509 and No.JCYJ20150617155357681), PRC.

REFERENCES

- [1] A. Afanasyev, I. Moiseenko, and L. Zhang. ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005, NDN, October 2012.
- [2] S. Arianfar, T. Koponen, B. Raghavan, and S. Shenker. On preserving privacy in content-oriented networks. In *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking, ICN '11*, pages 19–24, New York, NY, USA, 2011. ACM.
- [3] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC)*, 9(1):1–30, 2006.
- [4] A. Broder and M. Mitzenmacher. Network applications of bloom filters: A survey. *Internet mathematics*, 1(4):485–509, 2004.
- [5] S. Burnett, N. Feamster, and S. Vempala. Chipping away at censorship firewalls with user-generated content. In *USENIX Security Symposium*, pages 463–468. Washington, DC, 2010.

- [6] A. Chaabane, E. De Cristofaro, M. A. Kaafar, and E. Uzun. Privacy in content-oriented networking: Threats and countermeasures. *SIGCOMM Comput. Commun. Rev.*, 43(3):25–33, July 2013.
- [7] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Designing Privacy Enhancing Technologies*, pages 46–66. Springer, 2001.
- [8] X. Cui, L. C. Hui, S. Yiu, and Y. H. Tsang. Study of censorship in named data networking. *Advanced Multimedia and Ubiquitous Engineering: Future Information Technology*, 2:145, 2015.
- [9] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. Technical report, DTIC Document, 2004.
- [10] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. R. Karger. Infranet: Circumventing web censorship and surveillance. In *USENIX Security Symposium*, pages 247–262, 2002.
- [11] C. S. Leberknight, M. Chiang, H. V. Poor, and F. Wong. A taxonomy of internet censorship and anti-censorship, 2012.
- [12] E. Massawe, S. Du, and H. Zhu. A scalable and privacy-preserving named data networking architecture based on bloom filters. In *Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on*, pages 22–26, July 2013.
- [13] A. Stubblefield and D. S. Wallach. Dagster: censorship-resistant publishing without replication. *Rice University, Dept. of Computer Science, Tech. Rep. TR01-380*, 2001.
- [14] E. Uzun, S. DiBenedetto, G. Tsudik, and P. Gasti. Anonymous named data networking application. In *CCNxCon, CCN Community Meeting*, 2011.
- [15] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman. Telex: Anticensorship in the network infrastructure. In *USENIX Security Symposium*, 2011.
- [16] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, et al. Named data networking (ndn) project. *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

Xingmin Cui Xingmin Cui (S'15) received her B.Eng degree in computer science from Shandong University in China in 2012. She is currently working toward the Ph.D. degree in the Department of Computer Science, The University of Hong Kong. Her research interests include mobile security, program analysis and named data network (NDN) security.

Yu Hin Tsang Yu Hin Tsang received his B.Eng degree in computer science from The University of Hong Kong in 2012. He is currently a part-time MPhil student in the Department of Computer Science, The University of Hong Kong.

Lucas C.K.Hui Lucas C. K. Hui (M'93-SM'03) is the founder and Honorary Director of the Center for Information Security and Cryptography (CISC), and an associate professor in the Department of Computer Science, The University of Hong Kong. He received the B.Sc. and M.Phil. degrees in computer science from The University of Hong Kong, and the M.Sc. and Ph.D. degrees in computer science from the University of California, Davis. His research interests include different and diversified areas in information security, including authentication services, privacy protection in e-commerce, mobile network security, smart grid security, security system design involving hardware and software, computer forensics, privacy and security in e-learning systems, and cryptography. He has published more than 100 internationally referred research papers. He is the Principal Investigator of several applied research projects relating to IT security with a total sum of around US\$3 millions. He has been appointed by the Hong Kong SAR Government as a member of the Advisory Committee on Code of Practice for Recognized Certification Authorities for a period of two years from 1 February 2011 to 31 January 2013. In summary, Dr. Hui had extensive experiences in carrying out interdisciplinary research projects containing information security component.

S.M.Yiu S. M. Yiu (A'90-M'99) is currently an associate professor of Department of Computer Science, The University of Hong Kong. He has been working in the research area of cryptography and security for more than 10 years. Besides security, he also works in bioinformatics.

Bo Luo Bo Luo is an undergraduate student in Beijing University of Posts and Telecommunications, China. She participated in the work described in this paper when she worked as a summer intern in the Department of Computer Science, The University of Hong Kong.