This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing

Achieving Content-Oriented Anonymity with CRISP

Peng Zhang, Qi Li, and Patrick P. C. Lee

Abstract—As a popular realization of Information-Centric Network (ICN), Named Data Networking (NDN) greatly improves the efficiency of Internet content-distribution. A feature of NDN is that it improves privacy, as no addresses are needed for either the content consumer or publisher. However, NDN packets contain content names, and hence a well-motivated adversary can still deduce *what* content the user is requesting once it can link the packets and users. How to provide privacy in NDN, given its unique data retrieval mode, is an open problem. In this paper, we explore a specific content-oriented anonymity model called *content-user unlinkability*, which breaks the relationship between the content and the requesting user. We argue that achieving content-user unlinkability efficiently is a non-trivial task, since existing tunnel-based approaches will largely dismiss content caching of NDN, resulting in large content retrieval delay. To this end, we propose CRISP, namely Cooperative Random IntereSt Propagation. In CRISP, routers cooperate to form full-meshed groups, within which content requests are randomly propagated before they are forwarded to content producers. We show CRISP can achieve probable content-user unlinkability with probabilistic models. Extensive simulations demonstrate that CRISP outperforms existing solutions including ANDANA and Crowds, in terms of both content retrieval latency and data throughput.

Keywords—Named Data Networking, anonymity, content-user unlinkability

1 Introduction

Content distribution has become a dominant source of today's Internet traffic [1], [2]. However, the Internet was designed to provide "connection-based" services, rather than "content-based" services. To address this issue, Named Data Networking (NDN) is proposed [3], [4]. NDN identifies each piece of content with a globally-unique, location-independent *name*. Users request content by name, rather than by location. Also, content can be *cached* within the network, so as to allow efficient content delivery.

In addition to the enhanced design for content distribution, NDN also offers some privacy benefits. Specifically, NDN uses a name-based transmission model, which eliminates both source/destination addresses from packets. Thus, it is difficult for an adversary to deduce *who* is requesting a content by inspecting on packet headers. Also, since routers in NDN have caches, a request for specific content can be satisfied by routers caching the content, without going all the way to content publishers. Thus, an adversary (e.g., a central censorship authority) may not even observe the content requests.

Unfortunately, NDN raises new privacy threats as well. For one reason, NDN relies on content names to route packets. Compared with plain IP addresses, these names bear more meaningful information about content. Thus, an adversary may know exactly *what* content the packets are carrying [5]. For another reason, in-network caching also leads to information leakage: a local adversary can deduce whether its neighbors have accessed some (sensitive) content by monitoring an NDN router's response time of that content [6].

In sum, privacy issues in NDN are inherently different from those in traditional IP networks. In traditional IP networks, an adversary knows *who* is requesting some content (by inspecting on packet addresses), but may not know *what* the content is being requested. On the other hand, in NDN, an adversary cannot easily find *who* is requesting some content, but it can infer exactly *what* content is requested (by inspecting on content names).

Given the distinct nature of privacy in NDN, anonymity models designed for traditional IP networks may no longer be appropriate. Most existing anonymity models are targeted at unobservability, sender/receiver anonymity, and sender-receiver unlinkability [7]. Of these anonymity goals, sender-receiver anonymity can be achieved more efficiently. It means that one communication endpoint that sends packets (sender), and another endpoint that receives packets (receiver), should not be linked in the same session. In NDN, however, content retrieval proceeds in a recursive way: each router fetches content on behalf of its last-hop router. That is, packets are forwarded hop-by-hop, without any notion of end-to-end communication.

We believe that there is a need to design new anonymity models for NDN. Without such a model, existing anonymity schemes designed for NDN are still rooted in the traditional philosophy of building tunnels like Tor [8]. However, tunnels will totally dismiss universal content caching, a key feature of NDN for better supporting content distribution [9].

This paper explores a specific content-oriented

Peng Zhang is with Dept. of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China (E-mail: p-zhang@xjtu.edu.cn).
Qi Li is with Graduate School at Shenzhen, Tsinghua University, Shenzhen, China (E-mail: liqi@csnet1.cs.tsinghua.edu.cn).
Patrick P. C. Lee is with Dept. of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong (E-mail: pclee@cse.cuhk.edu.hk).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing

anonymity model termed *content-user unlinkability*. The anonymity goal of the model is to break the relationship between content and the requesting user in communication traffic. Under this model, we present Cooperative Random IntereSt Propagation (CRISP), a new content-oriented anonymity protocol for NDN that builds on the content caching feature of NDN. In CRISP, NDN routers randomly propagate request packets (termed *Interests* in NDN) among themselves, so as to hide which router an Interest comes from. Then, even if a compromised router knows what content is requested by inspecting the content name carried in the Interest, it cannot deduce from which router originates the Interest, and hence cannot infer which user requests the content. One important feature of CRISP is that the random propagation process works seamlessly with the standard Interest propagation process. For this reason, CRISP can build on existing NDN data structures (e.g., PIT), thereby making it readily deployable atop NDN.

Our idea of randomly propagating Interests is inspired by Crowds [10], an anonymity protocol for web transactions in IP networks. However, CRISP distinguishes itself from Crowds in two key aspects. First, CRISP leverages caching in NDN for low-latency content retrieval: any router that caches the content can satisfy the request, and hence the request does not need to traverse more random hops. In this sense, CRISP can be seen as a variant of Crowds that enables caching among crowd members. Second, CRISP introduces multiple-layer random propagation for improved anonymity, which can be seen as an extension to Crowds (which uses single-layer random propagation). As a result, CRISP improves both performance and anonymity over Crowds. We formally analyze and evaluate the improvements in Sections 4 and 5.

In summary, we make the following contributions:

- We introduce a new content-oriented anonymity model for NDN called *content-user unlinkability*. To the best of our knowledge, our work presents the first study of modeling anonymity specifically for content-oriented networks.
- We propose CRISP, a new anonymity protocol for NDN that supports content caching for lower content retrieval latency. To analyze the performance and anonymity achieved by CRISP, we propose new mathematical models based on discrete-time Markov chains, which are more powerful than the simple probability model in Crowds [10].
- We implement CRISP in ndnSIM [11], and demonstrate that it outperforms existing approaches including ANDANA and Crowds, in terms of content retrieval latency and data throughput.

The rest of the paper proceeds as follows. Section 2 presents the content-oriented anonymity model in NDN. Section 3 introduces the idea, design, and implementation of CRISP. Section 4 formally analyzes the anonymity and performance of CRISP based on discrete-time Markov chains. Section 5 presents simulation results based on ndnSIM. Section 6 reviews related work, and Section 7 concludes.

2 Problem Statement

This section formulates the content-oriented anonymity problem. We first give a brief overview of NDN, and then specify the adversarial model and anonymity model used throughout this paper.

2.1 NDN Overview

This paper focuses on a concrete content-oriented network based on Named Data Networking (NDN) [3]. NDN communication builds on a set of *content routers* (or *routers* for short), which forward content over the network and interconnect consumers (i.e., users that request content) and publishers (i.e., servers that publish content). Without loss of generality, we assume routers are organized in multiple *layers*, as shown in Fig. 1. The bottom layer of routers that directly connect with users are called *access routers* (i.e., Layer 1 routers in Fig. 1). Such layered deployment of NDN follows typical IP networks, and has also been adopted in previous works [12]).

There are two types of packets in NDN: *Interest* and *Data*. Each user requests a piece of content by sending an Interest to an access router, which then propagates the Interest in the network. When a router receives an Interest, it first checks its local cache. If the requested content is cached, the router returns a Data encapsulating the content. As shown in Fig. 1, the Interest sent by user U is forwarded by $R3$ and $R2$, and then arrives at $R1$, which has the content in its cache. $R1$ returns a Data without propagating the Interest to the publisher. The Data will then be cached by $R2$ and $R3$, so that they can serve the same Interests later. This in-network caching feature of NDN saves the bandwidth of forwarding redundant content, and makes content distribution more efficient.

In addition to content caches, each router also keeps two data structures: the *Forwarding Information Base (FIB)* and *Pending Interest Table (PIT)*. The FIB keeps a list of the router's outgoing *faces* (i.e., generalized interfaces that connect to a network or an application), and is used to make forwarding decisions for Interests. It resembles a traditional IP routing table, with the difference that the FIB lookup is based on content names rather than destination addresses. On the other hand, the PIT keeps track of all incoming faces of pending Interests. When a router receives an Interest, it inserts a record to its PIT; when it receives a Data packet, it checks its PIT to decide which face to forward the Data. The router may aggregate multiple Interests for the same content as a single PIT entry that contains a list of incoming faces, so as to forward the returned Data packets over all these faces.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing

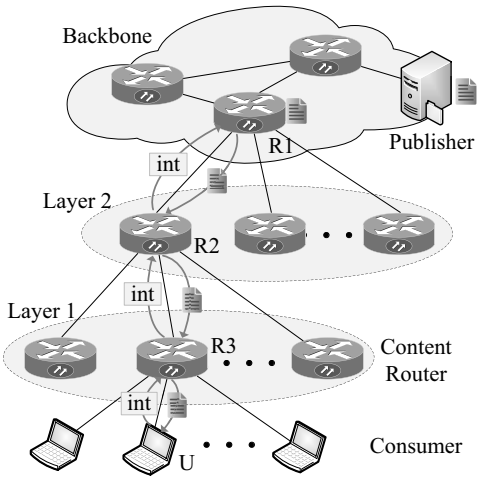


Fig. 1. A simple layered structure for NDN.

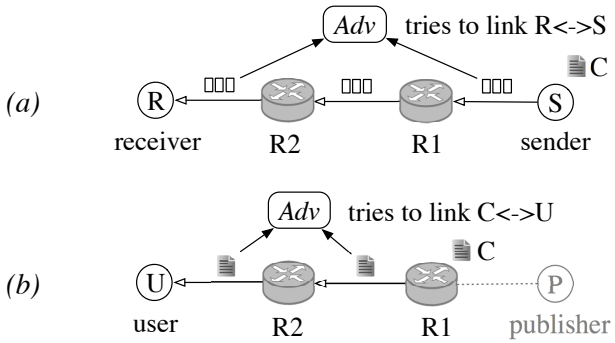


Fig. 2. A comparison of (a) sender-receiver unlinkability and (b) content-user unlinkability. In (a), an end-to-end connection is established between the sender S and receiver R, and the adversary is aimed at linking S and R. In (b), a user U issues a request for some content C, which is fetched by routers in hop-by-hop way; the adversary is aimed at linking content C and user U.

2.2 Adversarial Model

We consider an adversary, denoted by Adv, that is *local* and *passive*. By local, we assume that Adv can only compromise a limited set of c routers in a network, and intercept the traffic passing through these routers. By passive, we assume that Adv is "honest but curious", such that it only passively monitors the intercepted traffic, but does not drop, inject, or modify any packets.

The goal of Adv is to monitor Interests (i.e., requests for content), and on observing an Interest for some sensitive content, try to deduce *who* has sent the Interest. We assume Adv has not compromised the access router of the user who sent the Interests, i.e., the user's access router is not among the c compromised routers. Otherwise, the user is immediately exposed as Adv can monitor all traffic from the user. This assumption is reasonable if c is much less than the number of routers in the network, and it is very unlikely that Adv selects the right access router of the user, without even knowing who the user is.

Noted above, we are modeling an adversary that only compromises a small number of routers (e.g., aggregate or core routers), rather than an ISP-wide adversary that can monitor all access routers. The reason that we do not consider the ISP-wide adversary is explained as follows.

First, an important advantage of NDN over traditional IP network is that NDN can leverage in-network caching to improve content delivery performance. If we assume all edge/access routers in NDN are nontrusted, then users should build tunnels through these routers (i.e., using encryption as in Tor [9]). However, previous work [12] also showed that caching at network edges can achieve most of the benefits brought by caching in NDN (the optimal caching strategy can only improve 17% over simple edge-caching).

Secondly, even if we use encryption-based anonymity mechanisms, like Tor, it can still be defeated by a global adversary. Moreover, recent studies even showed that a local adversary can also deanonymize Tor users [13], [14]. For instance, Ling et al. [14] showed that by compromising even a small number of entry and exit ORs (Onion Routers), an adversary can discover the sender-receiver relationship in Tor.

2.3 Anonymity Model

Based on the adversarial model, we propose a new content-oriented anonymity model. To elaborate, Fig. 2(a) shows the case of traditional IP networks, where a session/connection is set up between the *sender* and *receiver*. Even if packets can be encrypted (e.g., via SSL/TLS), the adversary can still discover this end-to-end connection via traffic correlation. The anonymity goal in most state-of-the-art techniques is to achieve the *sender-receiver unlinkability* [7], which breaks the communication relationship of a sender and the corresponding receiver. That is, even though the adversary can observe the communication of senders and receivers, it cannot infer if a given sender and a given receiver are related in the same communication session. There are other anonymity goals, such as sender anonymity, receiver anonymity, and unobservability [7]. However, sender-receiver unlinkability receives the most attention since it can be efficiently achieved with relatively low latency, as proven by the success of Tor [8].

In contrast to traditional IP networks, Fig. 2(b) shows the case of content-oriented networks, where a *user* sends a request for some content published by a *publisher*. Routers will fetch, cache, and return content for the user, and no end-to-end connection exists between the user and the publisher. Thus, it is difficult for the adversary to link the user and publisher. On the other hand, anonymity remains at risk since the traffic carries content with semantic names, rather than plain packets (which can even be encrypted). The adversary can inspect the content name in the traffic, and correlate the content with the user who requests it. Thus, our anonymity goal is to achieve *content-user unlinkability*. In a nutshell, we aim

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing

to decouple the relationship between content and the requesting user in the traffic. In the following, we give a formal definition of content-user unlinkability.

Definition 1: Content-User Unlinkability. Consider a user U issues an Interest for some content C via her access router R, and the Interest is observed by the adversary Adv. Then, Adv first guesses $\hat{R}$ as the user's access router, and continues to guess $\hat{U}$ as the user. Let Δ be the probability that $\hat{R} = R$. Then, we say the content C and user U are unlinkable with parameter Δ. □

As our adversarial model assumes that Adv does not compromise R (see Section 2.2), every user that uses R as the access router appears equally to be the one that issues the Interest. Thus, the probability that $\hat{U} = U$, i.e., Adv identifies U, is actually $\frac{\Delta}{|\mathcal{U}(R)|}$, where $\mathcal{U}(R)$ is the set of users that use R as the access router. However, here we merely use Δ to quantify content-user unlinkability, since the factor $\frac{1}{|\mathcal{U}(R)|}$ may vary across routers, depending on how many end hosts are connected.

According to Definition 1, Δ quantifies the protection level of content-user unlinkability, and $1/n \leq \Delta \leq 1$, where n is the number of all routers in the system. $\Delta = 1$ means no protection at all, i.e., the access router R is exposed to Adv; $\Delta = 1/n$ means highest protection, i.e., each router appears equally likely to be the access router. Between these two extreme points, $\Delta = 1/2$ quantifies an medium level of protection, meaning that from Adv's point of view, R appears no more likely to be than not to be the real access router. We give the following definition.

Definition 2: Probable Content-User Unlinkability. We say C and U has probable content-user unlinkability if $\Delta \leq 1/2$. □

In this paper, we view probable content-user unlinkability as a minimum anonymity requirement, and focus on how to achieve it in this paper.

Note that the current NDN design does not offer any content-user unlinkability since Adv can easily trace the Interest back to the access router, and then identify the user that sends the Interest. We will use Fig. 1 as an example to illustrate how this attack works.

Suppose Adv initially compromises a router ($R1$ in Fig. 1). On receiving an Interest for a sensitive content that Adv is interested with, it compromises the Interest's last hop ($R2$ in in Fig. 1). As a content is normally split into multiple Data packets, the user needs to send a sequence of Interests to retrieve the content. Thus, Adv can receive another Interest for that content, and continue to compromise the Interest's last hop ($R3$ in Fig. 1). This process continues until Adv finally compromises the access router of the user, monitors traffic from all users connected to this access router, and identifies the user that requests the content. As our system model assumes NDN routers are organized in layers, Adv only needs to compromise a small number of routers ($R1$, $R2$, $R3$ in Fig. 1) to succeed.

3 CRISP Design

This section presents CRISP, i.e., Cooperative Random IntereSt Propagation. CRISP aims for the following design goals:

- **Anonymity:** It provides probable content-user unlinkability (defined in Section 2.3) for any user retrieving a piece of content.
- **Performance:** It preserves the in-network caching feature of NDN so as to minimize the content retrieval latency.

In the following, we first introduce the basic idea of CRISP, and then present the protocol and implementation details.

3.1 Basic Idea

As already seen in Section 2.3, plain NDN offers no content-user unlinkability, and is vulnerable to Interest traceback. The core reason is that a compromised router can be sure that an Interest's last hop is on the Interest's forwarding path. Thus, Adv only needs to compromise a small number of routers (less than Adv's capability c) to traceback to the access router and reveal the user.

Our basic idea is to let routers randomly propagate Interests among themselves before forwarding the Interests towards the content publisher, This will generate confusion to Adv, since a compromised router cannot tell whether an Interest's last hop is forwarding the Interest, or is just randomly propagating it.

We use Fig. 3 as an example to further illustrate our approach. Suppose routers 1 to 7 at Layer 1 are all connected with router A at Layer 2, and user 2 issues an Interest. Instead of forwarding the Interest to A at Layer 2, router 1 propagates the Interest along a random path (e.g., $1 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 7$), until the Interest is forwarded to A. Then, the Interest is randomly propagated by A at Layer 2, until it is forwarded to another router at Layer 3. In this way, A compromised router that receives an Interest cannot tell whether the sending router actually initiates the Interest or just forwards the Interest on behalf of another router. For example, router 3 cannot be sure whether router 1 initiates the Interest, or it is just randomly forwarding the Interest. It is thus difficult for a local adversary Adv (defined in Section 2.2) to traceback to the access router of the user.

Note that the random propagation of Interests in CRISP is similar to Crowds [10], in which a web user randomly propagates its web requests among a crowd in order to hide she is the real requester. A key difference is that CRISP supports content caching, such that any router that caches the content can directly return it. This avoids generating a long propagation path as in Crowds.

3.2 Protocol

The core of CRISP is a random Interest propagation engine that is designed to be fully compatible with NDN. Content routers in the same layer form *random*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing

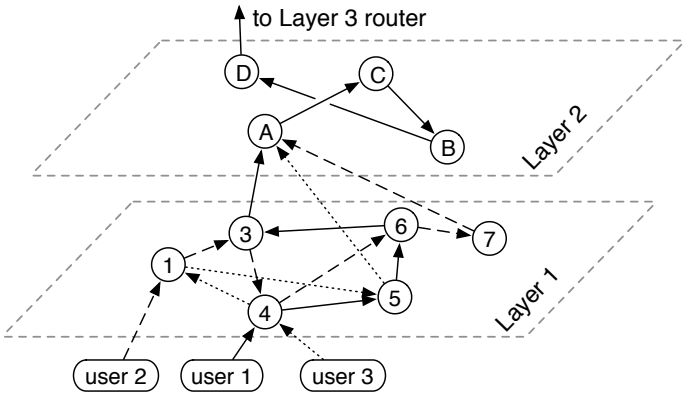


Fig. 3. An illustrative example of CRISP.

Algorithm 1: `CRISPProcessInterest(`int, f_{in}`)`

Input: int: Interest to be processed; f_{in}: the face from which int is received.
Data: $cached$: whether the content is cached, initialized to false.

```
1  /* Process Interests requiring no anonymity */
2  if int.flag = false then
3  |  NDNProcessInterest(int, f_in);
4  |  return;
5  end
6  /* Fast return: return the requested content if cached */
7  content ← CacheLookup(int.name);
8  if content then
9  |  cached ← true;
10 |  Send content to the incoming face f_in;
11 end
12 /* Process Interests from the lower-layer */
13 if f_in ∈ F_low and !cached then
14 |  Add an entry (int.name,f_in) into the PIT table;
15 |  Send int to a random face f_out ∈ F_crisp;
16 end
17 /* Process Interests from the same RPG */
18 if f_in ∈ F_crisp then
19 |  r ←R [0,1]; // generate a random number
20 |  if r ≤ p_f then
21 |  |  Add an entry (int.name,f_in) into the PIT table;
22 |  |  Send int to a random face f_out ∈ F_crisp;
23 |  else if !cached then
24 |  |  Add an entry (int.name,f_in) into the PIT table;
25 |  |  Send int to a face of upper layer f_out ∈ F_up;
26 |  end
27 end
28 return;
```

propagation groups (RPGs). Suppose that a router receives an Interest from a lower-layer router or an end user. If the requested content is not cached, then the router propagates the Interest to a randomly selected router in the same RPG. The chosen router flips a biased coin, and decides either to propagate the Interest to another random router in the RPG, or to submit it to an upper-layer router. In the following, we further elaborate the CRISP protocol, including how to process the Interest and Data packets in a router.

3.2.1 Interest Processing

Algorithm 1 summarizes the Interest processing protocol in CRISP. In the algorithm, F_{up} and F_{low} denote the sets of faces connected with upper-layer and lower-layer routers, respectively; F_{crisp} denotes the set of faces connected with routers in the same RPG. These three sets of faces can be configured at system bootstrap stage.

CRISP only operates on content that requires anonymity protection (Lines 1-5). An Interest has an anonymity flag, which is unset by default and is set by the user if the requested content needs anonymity. On receiving an Interest, the router checks whether the anonymity flag is set. If yes, the Interest will be handled by CRISP; otherwise, it will be processed normally by the underlying NDN protocol.

The first action of CRISP is called "fast return" (Lines 6-11): it first looks up the content cache with the Interest name. If the content is found, CRISP sends the content to the incoming face immediately. Then CRISP continues to process the Interest. Note that random propagation of the Interest is necessary even after the content has a cache hit, to maintain anonymity. The intuition is that if we stop propagating Interests after cache hits, the adversary would gain useful information. Consider if the content is very popular, a compromised router that receives an Interest can infer that the Interest is very likely to be initiated, rather than be forwarded, by the preceding router. We will present detailed analysis on the necessity of continuing random propagation in Section 4.2.

Lines 12-16 specify the case that the Interest is from a router at a lower layer. If the content is cached, nothing is performed as the content has just been returned (Line 10); otherwise, it creates an entry in the PIT to record the Interest name and its incoming face, and forwards the Interest to a randomly selected router in the same RPG. Here, the PIT keeps track of the random forwarding path, so that Data packets can be returned to the requesting user.

Lines 17-27 specify the case that the Interest is from a router in the same RPG. The algorithm flips a biased coin with "head" occurring with probability p_f, which we term *propagating probability*. If the result is "head", the Interest will be propagated to a randomly selected router in the RPG; otherwise, the Interest will exit the RPG, i.e., if the content is not cached, the router creates an entry in the PIT, looks up its FIB, and forwards the Interest to a router in the upper layer. If CRISP is enabled at this layer, routers will continue to randomly propagate the Interest according to Algorithm 1. Otherwise, routers can use other routing algorithms to forward Interests.

Two issues deserve further explanation in Algorithm 1.

Special treatment of layers. According to Algorithm 1, when a router receives an Interest from a lower-layer router, and it holds the corresponding Data, then it will not propagate the Interest. We explain this special treatment as follows. Suppose a user, say Alice, sends

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing

an Interest to her access router R which holds the Data. Then there are two possibilities: (1) R is compromised. In this case, Alice is exposed directly, and further random propagation is not necessary. (2) R is not compromised. In this case, R does not need to propagate the Interest, as the Interest has not been observed by the adversary yet, while propagating the Interest will only make the adversary observe it. The above reason applies when Alice is a lower-layer router and R is a high-layer router.

The anonymity flag. The anonymity flag can serve to amortize the system overhead of CRISP: users who do not care about privacy (at least not as much as performance), can just forego CRISP by not setting the flag. For users who care about privacy, they should set the anonymity flag for all Interests, no matter the Interests are requesting for sensitive contents or not. Otherwise, if they only set the anonymity flag for sensitive content, the adversary may deanonymize the user through the non-sensitive Interests (which are not following CRISP), and correlate the sensitive Interests with non-sensitive ones.

3.2.2 Data Processing

For Data, we do not distinguish whether they are for plain NDN, or for CRISP. Actually, a Data packet does not carry any anonymity flag as an Interest packet. Data packets are simply forwarded back to the requesting user, based on the information in the PIT, as in the normal NDN protocol. This implies that we keep the Data processing logic in NDN unchanged.

3.3 Static Paths against Interest Correlation

In the following, we discuss a special attack targeted at CRISP, and present a possible remedy approach.

Recall that in NDN, a large file must be decomposed into multiple Data packets, and a user should issue Interests for all these Data packets to retrieve the file. CRISP guarantees a single Interest cannot be linked with the user that sends it. However, if an adversary can correlate multiple Interests of the same file, it can identify the user with high probability. The reason is that when there are multiple Interests for the same file, the number of times that the user's access router precedes $\mathcal{R}$'s (the first compromised router that observes an Interest) is much higher than any other router in the RPG. This attack has been discussed in Crowds, and the authors mitigate this attack by letting each crowd member maintain a static path for a sufficiently long period (typically 24 hours).

CRISP can use the same approach, by letting each router maintain a static forwarding path for a sufficiently long period. However, this may lead to unbalanced load among routers. To address this problem, we propose to let access routers maintain a static path for each user. When a new user connects to an access router, the router establishes a new random forwarding path for the user, and will keep using this path for a long period.

To establish such a forwarding path, the access router chooses a random path identifier, and sends an Interest tagged with the path identifier. Each router processes this Interest according to Algorithm 1, with the difference that the router should choose a new path identifier (denoted as PID_{new}), and replace the path identifier carried by the Interest (denoted as PID_{old}) with PID_{new}. In addition, after selecting a random router (denoted as R) as the next hop, the router records this selection by creating an entry (PID_{new}, R) with index PID_{old} and inserting the entry into a hash table. Later on, when the same user sends Interests for content, routers will deterministically propagate these Interests along the forwarding path using the information kept in their hash tables.

Note that static paths are maintained on the router level, rather than the user level. According to Theorem 7.1 in [10], the number of paths that a router appears on (i.e., the number of PIDs that it needs to store) is of order $\boldsymbol{O}\left(\frac{1}{(1-p_f)^2}(1+\frac{1}{n})\right)$, where n is the number of routers in the RPG. Thus, a router only needs to store a small number of PIDs.

3.4 Implementation Details

We discuss some implementation details of CRISP in practical deployment.

3.4.1 Communication Among CRISP Routers

In CRISP, routers in the same RPG should propagate Interests among themselves. Also, each router in one RPG should be able to forward Interests to some upper-layer router. One way to enable such communication is to maintain physical connections among routers. That is, routers in the same RPG should be wired to form a full-mesh topology, and each router should be wired to another upper-layer router. This approach makes it very inefficient to make changes to the topology. As another option, routers can maintain persistent TCP connections with one another. This can be viewed as an overlay solution atop TCP/IP. A drawback is that the implementation is tied to the legacy TCP/IP protocol suite.

Instead of maintaining connections among routers, we use the name-based addressing of NDN for a better implementation. We illustrate how to maintain communication channels among all routers in the same RPG. Communication channels among lower-layer routers and upper-layer routers can be established in a similar way.

In our approach, routers in the same RPG announce the same prefix, say `/crisp/rpg1`, and each router also has a router-specific sub-prefix, say `/crisp/rpg1/R1`. When a router $R1$ needs to send an Interest/Data packet pkt with name $name$ to another router $R2$ in the same RPG `rpg1`, it encapsulates $name$ inside the prefix `/crisp/rpg1/R2/R1` to create a new name `/crisp/rpg1/R2/R1/name`. Then, $R1$ sends pkt with this new name, and pkt is routed towards $R2$. On

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing

receiving the packet, $R2$ realizes the packet is destined for it, and peels the outermost layer of the name to obtain $name$. In this way, $R1$ can build a communication channel towards $R2$. Similarly, the $R2$ can build a channel towards $R1$. Note that the implementation is transparent to legacy routers between CRISP routers, as legacy routers only forward the pkt as regular packet.

Note that each router needs to keep separate routing entries for every other router in the same RPG. While this will not pose a scalability problem, since each router can only appear in one RPG, and each RPG has a relatively small number of routers (we expect this number to be less than 100 in CRISP). Thus, a router only needs to maintain 100 extra routing entries in its FIB. Considering an NDN router can hold up to 10 million routing entries [15], the overhead of routing entries installed by CRISP is not significant.

3.4.2 Encryption

To prevent local eavesdroppers from observing the names contained in Interest and Data packets, we require CRISP to encrypt communication among routers. Each pair of routers in the same RPG can share a symmetric key, with which any traffic between them is encrypted. Continuing the above example, suppose $R1$ and $R2$ share a symmetric key k. When $R1$ needs to send an Interest/Data packet pkt with name $name$ to another router $R2$, it uses the name $\texttt{/crisp/rpg1/R2/R1/}\{\texttt{name}\}_k$ instead, where $\{\texttt{name}\}_k$ is the symmetric encryption of $name$ with key k. There are many approaches for establishing symmetric keys, and the discussion is beyond the scope of this paper. Apart from content names, payload of Data packets, and meta data may also reveal the identity of the user. Thus, we also require that the packet payload and meta data are also encrypted.

3.4.3 Membership Management

We assume there is a central registry that manages the membership of routers that participate in a RPG. The registry maintains states for all participating routers, and allows routers to join or leave the RPG. Each router retrieves a list of members in the same RPG, and periodically receives updates from the registry. Since routers can be assumed to stable (except for cases of hardware failures or upgrades), there should not been frequent member joins and leaves in CRISP. The membership management is mostly focused on handling faults. Many existing protocols can be leveraged to accomplish the fault-tolerant design.

4 Analysis

This section presents an analysis of CRISP on its anonymity and performance, based on discrete-time Markov chains. For anonymity, we show how the content-user unlinkability parameter Δ (see Section 2.3) is related to system parameters including the propagating probability p_f, the cache hit probability at routers, etc. For performance, we evaluate the path length in random Interest propagation, which characterizes the transmission latency incurred by CRISP.

TABLE 1
Summary of key notations.

n	the number of routers in the RPG.
c	the number of compromised routers in the RPG.
p_f	the propagating probability.
p_h	the cache hit probability in the network.
Δ	the identifying probability of the access router.
I	the event that the adversary identifies the user's access router.
$\mathcal{R}$	the first compromised router appearing on the Interest's forwarding path.
H_i	the event that $\mathcal{R}$ occupies the ith hop in the single-layer model.
L	the number of layers in the multi-layer model.
$H_{i,j}$	the event that $\mathcal{R}$ occupies the jth hop of layer i in the multi-layer model.
Len	the length of an Interest's forwarding path.

4.1 Notations and Assumptions

Table 1 summarizes the major notations used in our analysis. Let p_h be the *cache hit probability*, meaning that each router in the RPG has the same probability of caching the requested content. By fixing p_h, we implicitly assume that if a router does not cache the content requested by an Interest, it will not cache the content throughout the life cycle of the Interest. This ensures that when the Interest is randomly propagated within the RPG, the cache hit probability of the requested content remains unchanged. Considering the short lifetime of an Interest, this assumption is reasonable. In the following, we will also term the access router of the user that sends an Interest as the *initiating router* of the Interest.

4.2 Anonymity

In this subsection, we show CRISP provides *probable content-user unlinkability*, i.e., the identifying probability of the access router $\Delta \leq 1/2$ (see Definition 2). We will first consider the single-layer case where there is only one RPG, and then extend the model to multiple layers with more than one RPGs.

4.2.1 Single-Layer Model

Suppose there is one RPG at Layer 1, and the RPG consists of n routers, of which c routers are compromised by the adversary.

We assume the adversary launches the *predecessor attack* to identify the access router. The predecessor attack was originally developed to attack Crowds [10], and the attack is later extended in [16] to attack general anonymity schemes, including Onion Routing, Mix-Net, and DC-Net.

In the following, we give a brief introduction to the predecessor attack in the context of NDN. Assume there

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing

is at least one compromised router that receives a specific Interest, and let $\mathcal{R}$ be the first that receives it. That is, $\mathcal{R}$ is the first compromised router that appears on the propagating path of the Interest. In the predecessor attack, the adversary will always guess the predecessor of $\mathcal{R}$ as the access router (the router that initiates the Interest).

We explain why the predecessor of $\mathcal{R}$ is a good guess of the access router. Suppose a router compromised by Adv is not the first one that receives the Interest, then, the router that precedes it appears no more likely to be the access router than any other $(n-c)$ non-compromised routers. The success probability of guessing the access router is thus $1/(n-c)$. On the other hand, since $\mathcal{R}$ is the first one among all compromised routers that receive the Interest, then there is a non-zero probability that $\mathcal{R}$ is directly preceded by the access router. Conditioning on $\mathcal{R}$ is directly preceded by the access router, the success probability of the adversary is 1. Thus, the overall success probability when guessing the predecessor of $\mathcal{R}$ is:

$$\mathrm{P}(\mathcal{R} \text{ occupies 1st hop}) + \frac{\mathrm{P}(\mathcal{R} \text{ occupies non-1st hop})}{n-c},$$

which is strictly larger than $1/(n-c)$.

Formally, let I be the event that the adversary successfully identifies the initiating/access router. and H_i be the event that the first compromised router occupies the ith hop on the forwarding path. Let $H_{i+} = \bigcup_{j \geq i} H_j$. Then H_{1+} is the event that there is a least one compromised router on the forwarding path. Then, we can calculate the identifying probability Δ as:

$$\Delta = \mathrm{P}(I|H_{1+}) = \frac{\mathrm{P}(I)}{\mathrm{P}(H_{1+})} = \frac{\mathrm{P}(H_1) + \mathrm{P}(H_{2+})\frac{1}{n-c}}{\mathrm{P}(H_{1+})} \quad (1)$$

The steps of calculating $\mathrm{P}(H_i)$ are discussed in [10]. However, the calculation assumes there is no caching (i.e., $p_h = 0$) and thus cannot be used in our analysis. In the following, we will show a more general solution for $\mathrm{P}(H_i)$ where $p_h > 0$.

Recall in our design, we claim that it is necessary for a CRISP router to keep propagating an Interest even if there is a cache hit of the content. To analyze this, we consider the contrary case that when a CRISP router receives an Interest and has cached the content, it returns the content and stops propagating the Interest.

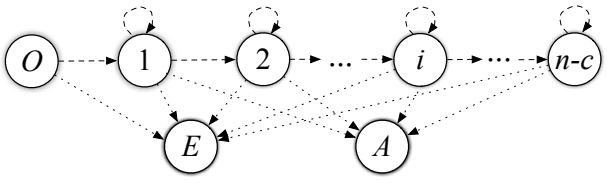


Fig. 4. The discrete-time Markov chain representing the number of different routers ever visited by a specific Interest. Here, state i means i different routers has been visited; O is the initial state; E and A are absorbing states.

We use a discrete-time Markov chain to capture the random propagation of an Interest in CRISP. In this Markov chain, each state i ($1 \leq i \leq n-c$) represents the state that i different routers have been visited. Apart from the above $n-c$ states, there are three additional states O, E, and A, which will be explained below.

A transition is triggered when the Interest is forwarded to a random router. Specifically, suppose that the chain is at state i, and a random router R is selected. If R is compromised, then the chain enters the absorbing state A, representing that the Interest is observed by the adversary. Now consider the case that R is not compromised. If R has the content, or R does not have the content but decides to stop random propagation and forward the Interest to an upper-layer router, the chain enters another absorbing state E. If none of the above two cases happens, R will propagate the Interest to another random router R'. If R' has already appeared on the path, then the chain stays at state i; otherwise, it enters state $i+1$. Finally, there is an initial state O representing that the user has just sent the Interest to the access router. After one step, the chain will either enter state 1 with probability $1-p_h$, or state E with probability p_h. The resulting Markov chain is shown in Fig. 4, and the transition probabilities are as follows:

$$\begin{cases} \mathrm{P}(O \to 1) = 1-p_h, \mathrm{P}(O \to E) = p_h \\ \mathrm{P}(i \to i) = \frac{i}{n} p_f, 1 \leq i \leq n-c \\ \mathrm{P}(i \to i+1) = \frac{n-c-i}{n}(1-p_h)p_f, 1 \leq i < n-c \\ \mathrm{P}(i \to E) = \frac{n-c}{n}(1-p_f + \frac{n-c-i}{n-c} p_f p_h), 1 \leq i \leq n-c \\ \mathrm{P}(i \to A) = \frac{c}{n}, 1 \leq i \leq n-c \end{cases}$$

With A being the absorbing state, the transition matrix can be represented as:

$$M = \begin{pmatrix} T & \boldsymbol{T_0} \\ \mathbf{0} & 1 \end{pmatrix}, \quad (2)$$

where T is the transition matrix of all states other than A, and $\boldsymbol{T_0}$ is the vector of transition probabilities into A. Then we have $T\mathbf{1} + \boldsymbol{T_0} = \mathbf{1}$, where $\mathbf{1}$ is the all 1's vector of length $n-c+2$. The distribution of the first time to absorbing state A follows discrete phase-type distribution $PH_d(T, \boldsymbol{\alpha})$:

$$\mathrm{P}(H_i) = \boldsymbol{\alpha}^T T^{i-1} T_0, \quad (3)$$

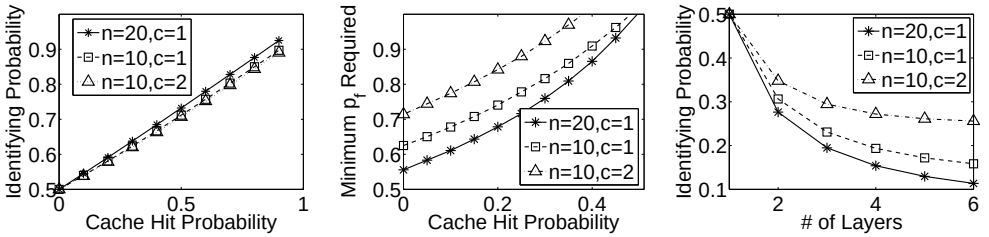
where $\boldsymbol{\alpha} = (1, 0, \ldots, 0)^T$ is the initial distribution for states $O, 1, 2, \ldots, n-c, E$. If we let $p_h = 0$, and use the resulting $\mathrm{P}(H_i)$ to solve Eq(1), we can obtain the same result as in [10]:

$$\Delta = \mathrm{P}(I|H_{1+}) = 1 - p_f \left(\frac{n-c-1}{n}\right) \quad (4)$$

Clearly, the minimum p_f to ensure $\Delta \leq 1/2$ when $p_h = 0$ is $n/(2(n-c-1))$. Using this p_f, we numerically solve the Markov chain for $\mathrm{P}(H_i)$ with $p_h > 0$, and use the results to calculate a new Δ, denoted as Δ'.

Fig. 5(a) shows the values of Δ' under different p_h's. We observe that Δ' exceeds $1/2$ when $p_h > 0$, and increases with p_h. This implies that the adversary can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing



(a) Identifying probability vs. cache hit probability (b) Minimum p_f required vs. cache hit probability (c) Identifying probability vs. # of layers

Fig. 5. Numerical results obtained from privacy analysis.

better identify the user if the requested content is popular. To ensure $\Delta' \leq 1/2$, we need to adjust p_f according to p_h. Fig. 5(b) shows the minimum p_f to ensure $\Delta' \leq 1/2$.

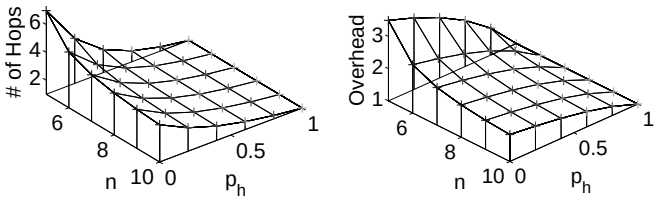
Normally, it is difficult to pre-determine the cache hit probability for a piece of content, and thereby adjust the value of p_f. Thus, CRISP lets routers keep propagating Interests even after cache hits. This means that p_h is set to zero, and we still use $p_f \geq n/(2(n-c-1))$, without being affected by different cache hit probabilities.

4.2.2 Multiple-Layer Model

Suppose there are $L > 1$ layers of routers, and routers of the same layer form a RPG. We are interested in whether anonymity is improved compared with the single-layer case. Recall that if an Interest goes out of the RPG at Layer $i-1$, it get received by another router at Layer i. We will term this router at Layer i as the *initiating router of Layer i*.

Without loss of generality, assume the RPG at each layer consists of n routers, of which c routers are compromised. Suppose that the first compromised router $\mathcal{R}$ occupies the jth hop at Layer i. We assume Adv launches what we call *iterative predecessor attack*, that is very similar with the predecessor attack defined in the single-layer model. First, Adv guesses $\mathcal{R}$'s predecessor, say R, as the initiating router of Layer i. Then, Adv further guesses the router that submits the Interest to R as the initiating router of Layer $i-1$. This process continues until the adversary guesses the initiating router of Layer 1. Δ is then the probability that the adversary succeeds in guessing the initiating router of Layer 1.

Then, we show how to calculate Δ in the multi-layer model. First, the probability that Adv succeeds in guessing the initiating router of Layer i is 1 for the case of $H_{i,1}$, and $1/(n-c)$ for the case of $H_{i,1+}$, just as in the single-layer model. For Layer k ($k < i$), since Adv did not observe the Interest at that layer, all the $n-c$ uncompromised router has equal probability to be the initiating router. That is, Adv has probability of $1/(n-c)$ to successfully guess the initiating router of Layer k ($k < l$). Thus, the identifying probability given $\mathcal{R}$ occupies the first hop at Layer i is $q_{i,1} = (1/(n-c))^{i-1}$, and that given $\mathcal{R}$ doesn't occupy the first hop is $q_{i,2} = (1/(n-c))^{i}$. Let $H_{i,j}$ be the event that $\mathcal{R}$ occupies the jth hop at Layer i. Take Fig. 3 for example, $H_{2,2}$ denotes the event that B is compromised. Also, we introduce two notations



(a) Expected path length (b) Bandwidth overhead

Fig. 6. Expected path length and bandwidth overhead, for different cache hit probability p_h and group size n.

$H_{i,j+} = \cup_{k \geq j} H_{i,k}$ and $H_{i+,j+} = \cup_{k \geq i} H_{k,j+}$. Then, we have:

$$\Delta = \mathrm{P}(I|H_{1+,1+}) = \frac{\sum_{i=1}^{L}[\mathrm{P}(H_{i,1})q_{i,1} + P(H_{i,2+})q_{i,2}]}{\sum_{i=1}^{L}\mathrm{P}(H_{i,1+})}, \tag{5}$$

where

$$\mathrm{P}(H_{i,j}) = \mathrm{P}(H_j)\left(1 - \mathrm{P}(H_{1+})\right)^{i-1}, \tag{6}$$

and $\mathrm{P}(H_j)$, $\mathrm{P}(H_{1+})$ take the values obtained in the single-layer model. Putting Eq(5) and Eq(6) together, we have:

$$\Delta = \mathrm{P}(I|H_{1+}) \cdot \frac{\mathrm{P}(H_{1+})\left(1 - \left(\frac{1-\mathrm{P}(H_{1+})}{n-c}\right)^{L}\right)}{\left(1 - \frac{1-\mathrm{P}(H_{1+})}{n-c}\right)\left(1 - (1-\mathrm{P}(H_{1+}))^{L}\right)} \tag{7}$$

By letting $\mathrm{P}(I|H_{1+}) = 1/2$, we report the identifying probability Δ in Fig. 5(c). We can see that with the increased number of layers, the value of Δ drops, that is, content-user unlinkability improves.

4.3 Performance

We then analyze the performance of CRISP, in terms of path length and bandwidth cost.

Path Length. The *path length* of an Interest is defined as the number of hops traversed by the Interest before it either leaves the RPG, or reaches a router that caches the content. Here, we leverage a modified version of the previous Markov chain in Section 4.2. The difference is we do not consider adversaries here, and thus there are n rather than $n-c$ normal states, with only two additional states O and E. The forwarding path length Len is then the number of transitions before the Markov chain enters the absorbing state E. The transition probabilities are as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing

follows.

$$\begin{cases} \mathrm{P}(O \to 1) = 1 - p_h, \mathrm{P}(O \to E) = p_h \\ \mathrm{P}(i \to i) = \frac{i}{n} p_f, 1 \le i \le n \\ \mathrm{P}(i \to i+1) = \frac{n-i}{n}(1-p_h)p_f, 1 \le i < n \\ \mathrm{P}(i \to E) = 1 - p_f + \frac{n-i}{n} p_f p_h, 1 \le i \le n \end{cases}$$

Similarly, the transition matrix for this Markov chain can be represented as:

$$\bar{M} = \begin{pmatrix} \bar{T} & \bar{\boldsymbol{T}}_{\mathbf{0}} \\ \mathbf{0} & 1 \end{pmatrix} \tag{8}$$

The distribution of the first time to absorbing state E follows the discrete phase-type distribution $PH_d(\bar{T}, \bar{\boldsymbol{\alpha}})$, where $\bar{\boldsymbol{\alpha}} = (1, 0, \ldots, 0)^T$ is the initial distribution vector of length $n+1$. Then, we have:

$$\mathbb{E}[Len] = \bar{\boldsymbol{\alpha}}^T (\mathcal{I} - \bar{T})^{-1} \mathbf{1}, \tag{9}$$

where $\mathcal{I}$ is the identity matrix and $\mathbf{1}$ is the all 1's vector.

We fix the number of colluding adversaries to $c = 1$, and report the results of Eq(9) in Fig. 6(a). We observe that the forwarding path length decreases when either the cache hit probability p_h or the group size n increases. This implies that the more popular the content is, the less hops the Interest traverses.

Bandwidth Overhead. The *bandwidth overhead* is defined as the ratio of extra bandwidth cost by CRISP on top of plain NDN. Let B_{crisp} be the bandwidth cost by CRISP to retrieve a Data packet, and let B_{ndn} be the bandwidth cost by plain NDN to retrieve the same Data packet. Then the bandwidth overhead of CRISP can be calculated as B_{crisp}/B_{ndn}. Let b be the unit bandwidth cost by a router to forward an Interest and the corresponding Data, we have:

$$B_{crisp} = \left(p_h + (1-p_h)(\frac{p_f}{1-p_f} + 2) \right) \times b$$

Here, the Data has probability of p_h to be cached by the access router, and thus the Interest does not need to be randomly propagated. In this case, a unit bandwidth cost is incurred. Otherwise if the Data is not cached, the Interest needs to be propagated $(\frac{p_h}{1-p_h} + 2)$ hops on average [10]. Similarly, we have $B_{ndn} = (p_h + 2(1-p_h)) \times b$.

We fix the number of colluding adversaries to $c = 1$, and report the bandwidth overhead in Fig. 6(b). We observe a similar trend as the forwarding path length in Fig. 6(a): the bandwidth overhead decreases with the increase of cache hit probability p_h and group size n.

5 Experiment

This section presents experimental results for CRISP, aiming to answer the following questions.

- Whether CRISP can achieve the probable content-user unlinkability when deployed on top of NDN (Section 5.2).
- How cache hit probability affects the performance of CRISP, in terms of transmission latency and bandwidth overhead (Section 5.3).
- Whether CRISP is more efficient for content retrieval compared with direct use of Crowds, and ANDANA (Section 5.4).
- How content popularity affects the performance of CRISP (Section 5.5).

5.1 Methodology and Setting

We implement CRISP with ndnSIM [11], an NDN module built on NS-3 [17]. We modify the Interest processing function in ndnSIM by intercepting the Interests with the anonymity flag set and passing it to the CRISP Interest processing function; the Data processing function is left unchanged.

To demonstrate the benefits of content caching offered by NDN, we also implement a variant of CRISP, named CRISP-NC (CRISP-No-Cache), for comparison. In CRISP-NC, when a router has cached the content requested by an Interest, it just keeps randomly propagating the Interest without returning the content. This mimics the direct usage of Crowds on top of NDN.

We also implement a simplified version of ANDANA [9] with ndnSIM. There are two flavors of ANDANA, one using Interest-based hybrid encryption, and the other using session-based symmetric encryption. Here, we choose the former, as the session-based approach has a problem that packets of the same session can be linked [9]. The hybrid encryption uses RSA-OAEP [18] to encrypt the symmetric key, and uses AES in CBC mode [19] to encrypt Interest with the symmetric key. The latter part (AES-CBC) is very fast, while the former part (RSA-OAEP) is relatively slow. We have implemented the above hybrid encryption with OpenSSL [20], and measure running time on a Linux desktop with an Intel quad-core 3.2GHz CPU and 4GB memory. We find that for a 1KB plaintext, the encryption and decryption operation cost 0.05ms and 0.42ms, respectively.

The topology used in our experiment consists of one consumer, one producer, and ten routers, connected by point-to-point links with uniform delays of 5ms. Here, we choose this relatively small topology mainly to validate our prototype, and conduct more controllable experiments. When experimenting with CRISP, we let these ten routers form a RPG; for experiments using ANDANA, we randomly select two routers as the anonymizing routers (one for entry and the other for exit). Since ANDANA works on overlay networks, the expected distance between any two anonymizing routers can be the diameter of the Internet. Thus, to mimic the overlay connections in ANDANA, we conservatively set the propagation delay among anonymizing routers to 30ms (6 hops, each with delay of 5ms).

To simulate a cache hit probability of p_h, we let each router retrieve and cache p_h of all Data packets randomly, in prior to experiments. We set $c = 2$, i.e., 2 out of the 10 routers are compromised. Then, p_f should be no less than $10/(2(10-2-1)) = 0.71$, according to Eq(4), and here we set $p_f = 0.8$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing

5.2 Probable Content-User Unlinkability

This experiment verifies whether CRISP achieves the content-user unlinkability defined in Section 2.3. We let the consumer issue 10^5 Interests using CRISP, and measure two metrics: (1) the observing ratio, the percentage of Interests that are received by least one compromised routers; (2) the identifying ratio, the percentage of Interests through which the adversary successfully guesses the consumer's access router. Thus, the identifying probability Δ defined in Eq(1) can be calculated as (1) divided by (2).

Fig. 7 (Left) and (Middle) show the observing ratio and the identifying ratio, respectively. There are two cache hit probabilities $p_h = 0.3, 0.5$, and for each p_h, points marked (a) represent the standard CRISP, and points marked (b) represent CRISP that stops random propagation after cache hits. From Fig. 7 (Left), we can see that the observing ratios increase when p_f increases, and (b) increases lower than (a). The reason is simple: when p_f is large, an Interest would traverse more hops on average, and thus is more likely to be received by the adversary; when routers stop random propagation after cache hits, the adversary is less likely to receive the Interests. Fig. 7 (Middle) exhibits a similar trend, while the increase of identifying ratios is negligible compared with that of the observing ratios.

Fig. 7 (Right) shows the identifying probability Δ, i.e., the identifying ratio divided by the observing ratio. The dashed line correspond to the theoretical values calculated using Eq(1). We can observe that the simulation results are close to the theoretical ones, and for both $p_h = 0.3(a)$ and $p_h = 0.5(a)$, Δ drops below 0.5 after p_f reaches around 0.7, meaning that probable content-user unlinkability is achieved regardless of p_h in CRISP. However, p_f needs to be larger than 0.9 to make $\Delta \leq 0.5$ for $p_h = 0.3(b)$, and there is even no feasible p_f for $p_h = 0.5(b)$. This indicates the necessity of keeping propagating Interests after cache hits in CRISP, confirming our analysis in Section 4.2.1.

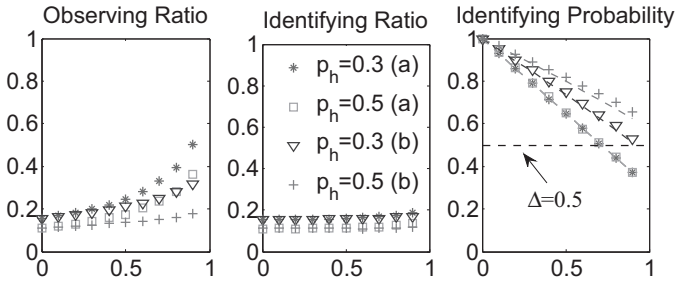


Fig. 7. The observing ratio, identifying ratio, and identifying probability for $p_h = 0.3$ and 0.5. (a) represents the standard CRISP, and (b) represents CRISP that stops random propagation after cache hits.

5.3 The Impact of Cache Hit Probability

This experiment evaluates the impact of cache hit probability on the performance of CRISP, in terms of Interests' forwarding path length, transmission delay, and overall bandwidth overhead. Similar to the previous experiment, we let the consumer issue 10^4 Interests at a constant rate of 100 Interests per second using CRISP. For comparison, we also repeat the experiment using plain NDN, ANDANA, and CRISP-NC.

5.3.1 Path Length

We first examine an Interest's forwarding path length, measured by the number of hops traversed by the Interest until it is either forwarded to the publisher or satisfied by a router along the path. Fig. 8(a) shows the expected forwarding path length for different cache hit probabilities in CRISP. The analytical results are obtained using Eq(9). We observe that the analytical results match the simulation results well. Now we take a closer look at the distribution of forwarding path length in Fig. 8(b). Here, CRISP-0.1 and CRISP-0.2 stands for CRISP with cache hit probability of 0.1 and 0.2, respectively. We observe that the path length of CRISP decreases when the cache hit probability increases from 0 to 0.2. As ANDANA only uses two anonymizing routers (one for entry, and one for exit), the path length is a constant of 3 hops, and is not shown in the figure.

5.3.2 Transmission Delay

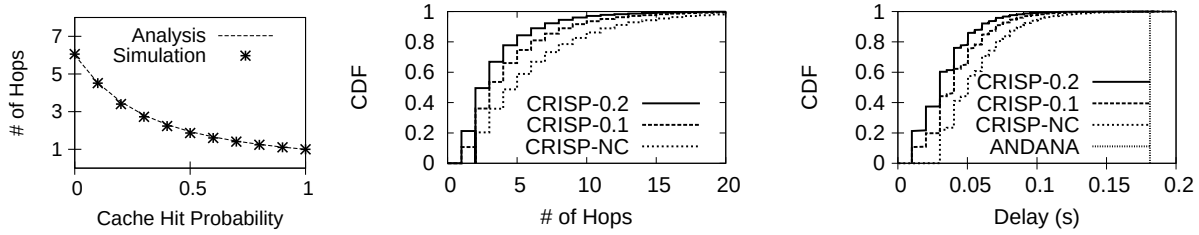
Here we examine the metric of per-Interest transmission delay, defined as the elapsed time between the instant that the user sends an Interest and that she receives the corresponding Data. Fig. 8(c) shows the cumulative distributions of per-Interest transmission delay for CRISP-0.1, CRISP-0.2, and CRISP-NC. The distributions are pretty similar to those in Fig. 8(b). This is because we have set the propagation delay of each link to 5 ms. The delay for ANDANA is roughly a constant of 180ms, mainly due to the packet propagation delay ($30\text{ms}\times 3 \times 2$). Since the traffic load is light, there is no queueing delay incurred by hybrid encryption/decryption functions. Note that this delay is higher than 20ms as reported in [9]. The reason is that the authors used a linear topology with four NDN nodes (one consumer, one producer, and two anonymizing routers), connected with point-to-point links. This simple topology may not correctly reflect the real scenarios, where anonymizing routers are distributed across multiple ISPs' networks.

5.3.3 Bandwidth Overhead

Finally, we examine the bandwidth overhead incurred by CRISP. We measure two metrics N_1 and N_2, i.e., the number of Data packets sent by all routers using CRISP and plain NDN, respectively. Fig. 9(a) reports the value of $N1/N2$, and also analytical results obtained in Section 4.3. The simulation results ($N1/N2$) agree with the analytical ones, indicating the bandwidth overhead drops with increased p_h.

Then, we look closer at the traffic load of each router in the RPG. Fig. 9(b) reports the amount of Data packets

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing

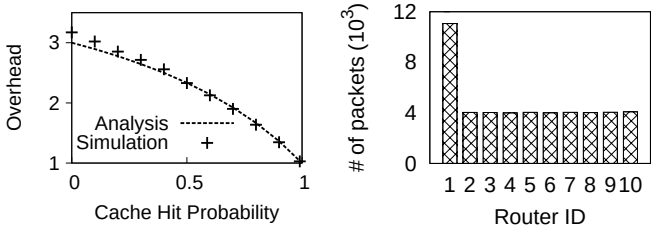


(a) Expected forwarding path length vs. cache hit probability

(b) Distribution of forwarding path length

(c) Distribution of per-Interest transmission delay.

Fig. 8. Experimental results demonstrating the impact of cache hit probability on path length and delay.



(a) Bandwidth overhead

(b) Number of packets received

Fig. 9. Experimental results demonstrating the impact of cache hit probability on bandwidth overhead.

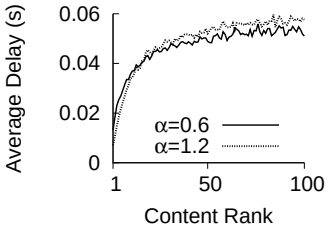
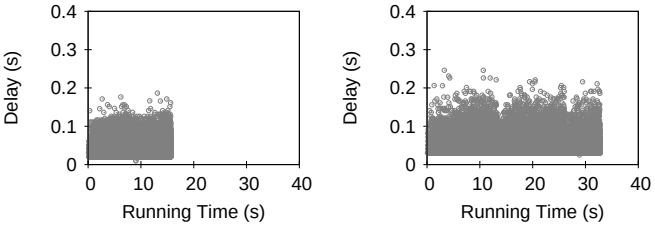


Fig. 12. Average per-Interest transmission delay. Content popularity conforms to the Zipf-Mandelbrot distribution with slop parameter $\alpha = 0.6$ and $\alpha = 1.2$.

received by each router when $p_h = 0.1$. We can observe that all routers have roughly the same traffic load, except for the first one, which is the access router. The reason is that the access router will receive all Data packets (except for cached ones).

5.4 The Efficiency of Content Retrieval



(a) CRISP

(b) CRISP-NC

Fig. 10. Per-Interest transmission delay when retrieving a file of 40MB using CRISP and CRISP-NC.

This experiment evaluates the content retrieval efficiency of CRISP, compared with CRISP-NC (i.e., direct use of Crowds), and ANDANA We let the consumer retrieve a file of 40MB from publisher anonymously with CRISP, CRISP-NC, and ANDANA, respectively. The performance metrics we consider include retrieval delay and data throughput. We use a simple sliding window at the consumer to adjust the sending rate of Interests, so as to saturate the network bandwidth. This is in contrast to former experiments, where the traffic rate is constant and low.

5.4.1 Content Retrieval Delay

Fig. 10 reports the content retrieval delay for CRISP and CRISP-NC, respectively. We can see that the retrieval delays of single Data packets in CRISP are lower than those in CRISP-NC, but the difference is not remarkable. However, CRISP finishes downloading the whole file in less than 20s, nearly half the time for CRISP-NC. This demonstrates the advantage of CRISP in better utilizing the network bandwidth with the help of caching.

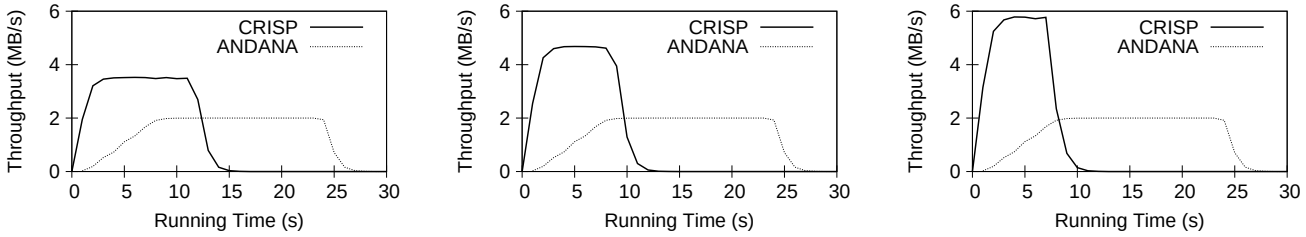
5.4.2 Data Throughput

We vary the link capacity from 30Mbps to 50Mbps, and measure the data throughput in CRISP and ANDANA. As shown in Fig. 11, CRISP can achieve a higher throughput with the increase of link capacity. On the other hand, ANDANA has a maximum throughput around 2MB/s, no matter how large the link capacity is. This is due to the fact that hybrid decryption for an Interest costs roughly 0.42ms (mainly due to the asymmetric decryption part), resulting in a maximum throughput around 2000 packets per second. As each packet has 1KB, the throughput is 2MB/s. The results show that the throughput of ANDANA may be limited by the computation capacity of routers, due to hybrid encryption/decryption. On the other hand, CRISP has no such limitation and can make full use of network bandwidth.

5.5 The Impact of Content Popularity

This experiment investigates the impact of content popularity on the performance of CRISP. We classify all Data packets into $N = 100$ ranks of popularity, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing



(a) Bandwidth=30Mbps (b) Bandwidth=40Mbps (c) Bandwidth=50Mbps

Fig. 11. Data throughput for CRISP and ANDANA when the link capacity is set to 30Mbps, 40Mbps, and 50Mbps.

let all routers constantly request these contents to fill their caches. The request ratios conform to the Zipf-Mandelbrot distribution [21]. In specific, the request ratio of rank-k content is $f(k) = \frac{1/(k+q)^{\alpha}}{H}$, where α is the slop parameter, q is a constant, and $H = \sum_{k=1}^{N} f(k)$ is a normalization factor. The cache size of each router is set to 10KB, meaning that it can hold up to 10 Data packets. For cache replacement, we use Least Recently Used (LRU), a simple yet near-optimal strategy [22].

We let one consumer retrieve these contents for $10,000$ seconds with CRISP, and report the average per-Interest transmission delay in Fig. 12. We can observe contents that are more popular have smaller per-Interest transmission delays. This indicates that CRISP incurs smaller delay when users are retrieving relatively popular contents.

6 Related Work

6.1 Privacy in Traditional IP Networks

The research of private communication stems from Chuam's Mix-Net [23], a protocol for sending and receiving emails anonymously over the Internet. Based on Mix-Net, Onion Routing [8], [24] are later proposed to anonymize a wider spectrum of applications, with lower latency. Both Mix-Net and Onion Routing use layered encryption to eliminate the packet correlations at participating nodes.

Crowds [10] is another anonymity scheme designed for web transactions without layered encryptions. A crowd is defined as a set of client processes (termed jondos) that need anonymous web browsing. When a user needs to issue a web request, she joins a crowd as a jondo, and sends the request to a randomly selected jondo in the same crowd. This jondo would flip a biased coin to decide whether to forward the request to another randomly selected jondo, or submit the request directly to the web server. This process continues until the request is finally submitted. There are some variants of Crowds. Hordes [25] uses multicast routing when to reduce transmission latency. However, multicast routing cannot be widely enabled because of the scalability issue. D-Crowds [26] generalizes Crowds to a TTL-based deterministic forwarding scheme.

6.2 Privacy in Content-Oriented Networks

There are some works addressing privacy concerns in NDN. Arianfar et al. [5] propose a scheme for anonymous content retrieval in NDN. To publish a file, the publisher first divides it into objects of equal size, and XORs them together with other objects of cover files to generate a large pool of chunks. Users interested in the file issues requests for certain chunks that are enough for her to decode the file. This scheme introduces high communication overheads to retrieve cover files, which may hurt the performance of content delivery.

Lauinger et al. [6] study the possible privacy breaches due to the ubiquitous caching in NDN. They identify the "request monitoring attack", which can be launched by adversaries connected to the same cache as the victim. Two countermeasures, namely selective caching and selective tunneling are suggested. This problem is later treated by [27], [28]. Mohaisen et al. [28] introduce random delays at edge routers when the requested content is cached, in order that adversaries cannot distinguish whether the content is in the cache of the edge router. Acs et al. [27] propose an approach to mask cache hits in content routers. In their approach, a random integer k is chosen according to a specific distribution, and cache hits are possible when the number of request reaches k.

Cryptography-based approaches were also applied to privacy protection in Information-Centric Networks [29], [30]. For example, Nabeel et al. use Paillier homomorphic cryptographic system to secure different messages in ICN [29]. Due to the usage of end-to-end encryption, these approaches will dismiss the benefits of content caches, and introduce relatively high computation overhead as well.

The scheme most relevant to ours is ANDANA [9], an anonymity protocol designed for NDN. It can be thought as an variant of Tor [8], that is adapted for NDN. Its basic idea is to build a tunnel using Interest packets. Each content router in the tunnel peels or adds a layer of encryption on each Interest passing through them. ANDANA has the following shortcomings: (1) it uses hybrid encryption which has a relatively high computation overhead, and thereby limit the users' throughput; (2) the tunnel-based approach can dismiss the benefits of universal caching in NDN; (3) encrypted data would be cached by content routers, but may never be requested again, resulting in wasted cache space.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2498603, IEEE Transactions on Dependable and Secure Computing

7 Conclusion and Future Work

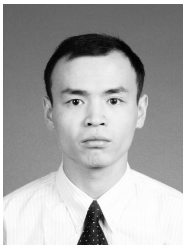
This paper introduced a new anonymity model named *content-user unlinkability*, which catches the basic requirement for anonymity in NDN, i.e., content should not be linked with the user requesting it. Under this model, we proposed CRISP, a new anonymous content retrieval scheme on top of NDN. We used a simple probabilistic model based on Markov chains to show CRISP can achieve the *probable content-user unlinkability* with a moderate bandwidth overhead. Experimental results showed that CRISP has a smaller content retrieval latency and larger data throughput, compared with other candidate schemes like Crowds and ANDANA.

The current paper is mainly focused on models and prototypes, aiming to demonstrate the applicability of adapting Crowds-like anonymization technique to NDN. Our future work includes building a full-fledged version of CRISP based on NDN Forwarding Demon (NFD [31]), and experimenting with it on real testbeds [32].

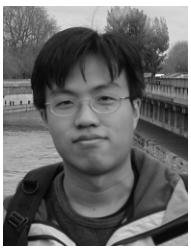
References

[1] Cisco Visual Networking Index, "Forecast and methodology, 2011-2016," *Cisco white paper*, 2012.
[2] Sandvine Intelligent Broadband Networks, "Global Internet Phenomena Report 2H 2012," 2012.
[3] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in *CoNEXT*, 2009.
[4] "The CCNx project," http://www.ccnx.org/.
[5] S. Arianfar, T. Koponen, B. Raghavan, and S. Shenker, "On preserving privacy in content-oriented networks," in *ICN*, 2011.
[6] T. Lauinger, N. Laoutaris, P. Rodriguez, T. Strufe, E. Biersack, and E. Kirda, "Privacy risks in named data networking: what is the cost of performance?" *ACM SIGCOMM CCR*, 2012.
[7] A. Pfitzmann and M. Hansen, "Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management – a consolidated proposal for terminology," 2008.
[8] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *USENIX Security*, 2004.
[9] S. DiBenedetto, P. Gasti, G. Tsudik, and E. Uzun, "ANDaNA: Anonymous named data networking application," in *NDSS'12*.
[10] M. Reiter and A. Rubin, "Crowds: Anonymity for web transactions," *ACM Transactions on Information and System Security*, vol. 1, no. 1, pp. 66–92, 1998.
[11] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," Technical Report. [Online]. Available: http://named-data.net/techreports.html
[12] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker, "Less pain, most of the gain: Incrementally deployable ICN," in *SIGCOMM*, 2013.
[13] S. Chakravarty, M. V. Barbera, G. Portokalidis, M. Polychronakis, and A. D. Keromytis, "On the effectiveness of traffic analysis against anonymity networks using flow records," in *PAM*, 2014.
[14] Z. Ling, J. Luo, W. Yu, X. Fu, D. Xuan, and W. Jia, "A new cell counter based attack against tor," in *ACM CCS*, 2009.
[15] Y. Wang, Y. Zu, T. Zhang, K. Peng, Q. Dong, B. Liu *et al.*, "Wire speed name lookup: A gpu-based approach." in *NSDI*, 2013.
[16] M. K. Wright, M. Adler, B. N. Levine, and C. Shields, "The predecessor attack: An analysis of a threat to anonymous communications systems," *ACM Transactions on Information and System Security*, vol. 7, no. 4, pp. 489–522, 2004.
[17] "The network simulator - NS-3," http://www.nsnam.org.
[18] M. Bellare and P. Rogaway, "Optimal asymmetric encryption," in *EUROCRYPT*, 1994.
[19] J. Daemen and V. Rijmen, *The design of Rijndael: AES-the advanced encryption standard*, 2002.
[20] "The OpenSSL project," http://www.openssl.org/.
[21] B. Mandelbrot, "Information theory and psycholinguistics: A theory of word frequencies." MIT Press, 1967.
[22] A. Sharma, A. Venkataramani, and R. K. Sitaraman, "Distributing content simplifies ISP traffic engineering," in *SIGMETRICS*, 2013.
[23] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.
[24] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected Areas in Communication*, vol. 16, no. 4, pp. 482–494, 1998.
[25] C. Shields and B. N. Levine, "A protocol for anonymous communication over the Internet," in *ACM CCS*, 2000.
[26] G. Danezis, C. Diaz, E. Käsper, and C. Troncoso, "The wisdom of Crowds: attacks and optimal constructions," in *ESORICS*, 2009.
[27] G. Acs, M. Conti, P. Gasti, C. Ghali, and G. Tsudik, "Cache privacy in named-data networking," in *IEEE ICDCS*, 2013.
[28] A. Mohaisen, X. Zhang, M. Schuchard, H. Xie, and Y. Kim, "Protecting access privacy of cached contents in information centric networks," in *AsiaCCS*, 2013.
[29] M. Nabeel and E. Bertino, "Efficient privacy preserving content based publish subscribe systems," in *SACMAT*, 2012.
[30] A. K. Maji and S. Bagchi, "*v*-CAPS: A confidentiality and anonymity preserving routing protocol for content-based publish-subscribe networks," in *SecureComm*, 2011.
[31] "NFD - Named Data Networking Forwarding Daemon," https://github.com/named-data/NFD.
[32] "The NDN testbed," http://named-data.net/ndn-testbed/.

Peng Zhang received the Ph.D. degree in Computer Science from Tsinghua University in 2013. He is now an assistant professor of the Department of Computer Science and Technology, Xi'an Jiaotong University, China. He has been a visiting student at The Chinese University of Hong Kong and Yale University. His research interests include network security and privacy, software-defined networks, and information-centric networks.



Qi Li received the Ph.D. degree from Tsinghua University. He is now an associate professor with Graduate School at Shenzhen, Tsinghua University. He is working on different projects in security and networking, and has worked at ETH Zurich, University of Texas at San Antonio, The Chinese University of Hong Kong, Chinese Academy of Sciences, and Intel. His research interests are various topics including security, Internet, and large scale distributed systems.



Patrick P. C. Lee received the B.Eng. degree (first-class honors) in Information Engineering from the Chinese University of Hong Kong in 2001, the M.Phil. degree in Computer Science and Engineering from the Chinese University of Hong Kong in 2003, and the Ph.D. degree in Computer Science from Columbia University in 2008. He is now an associate professor of the Department of Computer Science and Engineering at the Chinese University of Hong Kong. His research interests are in various applied/systems topics including storage systems, distributed systems and networks, operating systems, dependability, and security.