Title	Anonymity and Privacy in Named Data Networking based on Onion Routing and K-anonymity
Author(s)	北, 健太朗
Citation	大阪大学, 2023, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/91990
rights	
Note	

Anonymity and Privacy in Named Data Networking based on Onion Routing and K-anonymity

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2023

Kentaro KITA

List of Publications

Journal Papers

1. Kentaro Kita, Yuki Koizumi, Toru Hasegawa, Onur Ascigil and Ioannis Psaras, "Producer Anonymity based on Onion Routing in Named Data Networking," *IEEE Transactions on Network and Service Management (IEEE TNSM)*, vol. 18, no. 2, pp. 2420-2436, Jun. 2021.

2. Kentaro Kita, Yuki Koizumi and Toru Hasegawa, "Private Retrieval of Location-related Content Using k-anonymity and Application to ICN," *Computer Networks*, vol. 209, p. 108908, May. 2022.

Refereed Conference Papers

1. Kentaro Kita, Yoshiki Kurihara, Yuki Koizumi and Toru Hasegawa, "Location Privacy Protection with a Semi-honest Anonymizer in Information Centric Networking," in *Proceedings of ACM Conference on Information-Centric Networking*, *Full Paper Session*, pp.95-105, Sep. 2018.

Non-Refereed Technical Papers

1. Kentaro Kita, Kai Ryu, Yuki Koizumi and Toru Hasegawa, "A Study on Location Privacy for Multicast-Based Services," in *IEICE Technical Report*, vol. 117, no. 353, IN2017-63, pp. 103-108, Dec. 2017 (in Japanese).

2. Kentaro Kita, Yuki Koizumi and Toru Hasegawa, "Multicast Algorithm to Protect Location-Privacy against Continuous Queries," in *IEICE Communication Society Conference*, B-7-48, Mar. 2018 (in Japanese).

3. Kentaro Kita, Yuki Koizumi and Toru Hasegawa, “A Study on Indistinguishability of Users in Privacy Preserving Location-Based Services,” in *IEICE Communication Society Conference*, B-7-20, Sep. 2018 (in Japanese).

4. Kentaro Kita, Yoshiki Kurihara, Yuki Koizumi and Toru Hasegawa, “Privacy Protection against a Semi-honest Anonymizer for Location-Based Services,” in *IEICE Technical Report*, vol. 118, no. 359, IN2018-67, pp. 49-54, Dec. 2018 (in Japanese).

5. Kentaro Kita, Yuki Koizumi and Toru Hasegawa, “A k-anonymity based Scheme for Private Location-related Content Retrieval in NDN,” in *IEICE Technical Report*, vol. 119, no. 461, IN2019-108, pp. 177-182, Mar. 2020 (in Japanese).

6. Kentaro Kita, Yuki Koizumi and Toru Hasegawa, “A Study on a Design of an Anonymizer Run on an Untrusted Node to Protect Location Privacy,” in *IEICE Communication Society Conference*, B-7-13, Sep. 2020 (in Japanese).

Preface

Anonymity and privacy are fundamental rights of individuals and crucial for a free society. Informally, users are said to be anonymous if they can take actions without revealing their identities and private if they can decide how, when, and for what purposes their private information could be released to other parties. However, the wide use of network communications inevitably exposes users to many anonymity and privacy threats. Packets pass through various networks potentially managed or compromised by malicious parties that attempt to reveal who is communicating with whom and the content of messages from the packet headers and payloads. As pervasive monitoring, censorship, and collection of personal data by ISPs, governments, and tech companies become serious and realistic concerns, the importance of online anonymity and privacy has been emphasized.

Fortunately, various information-theoretical and cryptographical techniques have been proposed to provide anonymity and privacy on the current IP network. Owing to the emergence of next-generation network architectures, these studies have recently been revisited. Among the network architectures, the author focuses on Named Data Networking (NDN), which aims to achieve efficient content distribution. NDN carries each piece of content in a Data packet, which is identified and located with a unique content name and signed by its producer. A consumer can retrieve some specific content with an explicit request, called an Interest packet, that specifies the content name.

Although NDN provides end-to-end content security with per-packet signatures, it does not provide anonymity and privacy by default owing to information leakage from the following two components: human-readable content names and publicly verifiable signatures. First, NDN uses hierarchically-structured human-readable content names like URL. Each content name contains the globally routable name of its producer, called a producer name, as a prefix to route Interest packets toward the producer and the remaining suffix identifies specific content of the producer. Such content names reveal more information regarding what content is being retrieved/published and who retrieving/publishing some specific content than the IP addresses. Second, Data packets are inherently bound to their producers since Data packets carry signatures, which are publicly verifiable.

Adversaries can reveal who published some specific content from the signature.

The main goal of this thesis is to design two distinct systems that provide particular types of anonymity and privacy by preventing the information leakage from these components, augmenting anonymity and privacy in NDN to make it a serious candidate for the next-generation network architecture. The author's design is based on novel anonymity model to capture communication features specific to NDN and privacy model to complement privacy aspects not appropriately formalized in existing studies. These rigorous definitions allow the author to comprehensively analyze the levels of anonymity and privacy the proposed systems provide through theoretical and/or empirical ways.

In the first part of this thesis, the author focuses on producer anonymity, which means to conceal who publishes some specific content. The author presents ACPNDN (Anonymous Content Publishing in NDN), as the first NDN-based system to provide producer anonymity under a realistic adversary model. ACPNDN uses pseudonyms of producers and rendezvous points together with onion routing. Consumers can retrieve content from an anonymous producer by sending Interest packets specifying the pseudonym instead of the producer name through the rendezvous point. Producer anonymity is provided because the pseudonym carries no information regarding the producer's identity and signatures are produced with a public/private key pair securely bound to the pseudonym. The core design of ACPNDN is based on hidden service currently deployed over Tor; however, the author improves hidden service by taking advantage of a feature of NDN that Interest packets do not carry any information regarding their senders. Through theoretical and empirical analysis, the author shows that ACPNDN provides a level of anonymity comparable to hidden service with smaller latency and is more resilient against a type of traffic analysis attack, which aims to break anonymity by inspecting traffic patterns. These results shed light on the fact that it is feasible to design NDN-based anonymity systems that are more efficient and secure than existing IP-based anonymity systems although content names and signatures in NDN reveal considerable amount of private information to network adversaries.

In the second part of this thesis, the author focuses on location privacy on NDN-based IoT platforms. An important feature of these platforms is that a consumer can retrieve content directly from IoT devices installed at a location of their interest (LOI) by sending an Interest packet specifying the location name as the name prefix (i.e., producer name) and the type of desired content as the suffix. Different from producer anonymity, location privacy aims to conceal which producer (i.e., location) is being retrieved content by a specific consumer. For protecting location privacy of consumers, the names of consumers' LOIs and signatures produced by IoT devices carried in Interest

and Data packets must be concealed from network adversaries in an end-to-end way. To this end, the author designs a system, called PLCR (Private Location-related Content Retrieval), by leveraging k-anonymity of location. Like a series of existing studies, PLCR conceals an LOI in a location anonymity set, which consists of the LOI and other $k - 1$ dummy locations; however, PLCR has the following distinctive features: (1) PLCR generates location anonymity sets according to the requirements rigorously defined by using the notions of entropy and t-closeness to guarantee that adversaries can infer the LOI with a probability close to $1/k$ and the geographical information of the LOI gained by adversaries is minimized; and, (2) in contrast to the adversary model of the existing studies where the anonymizer is honest, PLCR protects location privacy under a more realistic adversary model where the anonymizer is a semi-honest adversary. Specifically, PLCR carefully leverages a private information retrieval (PIR) protocol and a locally differentially private (LDP) protocol to allow consumers to retrieve content from their LOIs while concealing the LOIs from the semi-honest anonymizer. In addition, to protect location privacy for continuous requests of a single consumer, PLCR also leverages the feature of NDN that Interest packets do not carry any information regarding their senders. PLCR itself is designed for the NDN-based IoT platforms; however, the requirements for each location anonymity set and design rationale to realize a semi-honest anonymizer are independent of the underlying network architecture and thus can be leveraged for location-based services deployed on the IP network.

Acknowledgments

This thesis would not have been complete without the support, guidance, and friendship of many people. I express my appreciation to them here.

First and foremost, I am deeply grateful for having Professor Toru Hasegawa as my Ph.D advisor. Through six years of his mentorship, I learned a lot from his consistent courage not to give up until he understands the heart of a problem. I particularly appreciated his willingness to patiently discuss how to improve my research ideas and manuscripts with me, no matter how poorly I explain something. I would like to thank the rest of my committee: Professor Masayuki Murata, Professor Takashi Watanabe, Professor Hirozumi Yamaguchi, and Professor Hideyuki Shimonishi for carefully reviewing this thesis and providing valuable comments. I would also like to thank the rest of my mentors: Associate Professor Yuki Koizumi and Assistant Professor Junji Takemasa for valuable suggestions and patient discussion to improve my research ideas. They were also important advisors about non-academic matters.

I would like to thank my collaborators outside Osaka University: Ioannis Psaras and Onur Ascigil. In the collaboration with these great researchers at University College London, I found research activities to be very stimulating and profound. Their valuable comments based on their deep understanding of computer networking and craft of research considerably improved my research achievements.

Joining the Information Sharing Platform Laboratory is the best fortune I could have during my pursuit of the Ph.D. I want to express my appreciation to all the members of the laboratory. Among them, special thanks to Nozomi Oda, Rie Maeda, and Tomoko Ueshima for their kind support that always let me immerse myself in research activities, and to Yoshiki Kurihara, Yohji Yamamoto, Yasunaga Murai, Yukiya Yamane, Kurbonov Ulugbek, Tatsuro Seno, Akio Yamagushi, Hiroki Masuda, and Yutaro Yoshinaka for all the enjoyable conversation and insightful discussion both on academic and non-academic matters. I am thankful for the opportunity to work with these great colleagues with various philosophies and backgrounds.

My friends reminded me about the importance of non-academic parts of life and gave me the passion to face academic matters. Among them, special thanks to Takumi Matsuda and Ryota Sawada for being great friends ever since we met.

Finally, I am most deeply grateful for the support and love given by my family. Undoubtedly, this work would not have been complete without them. I particularly thank my father, mother, sister, grandfather, and grandmother, who fostered my curiosity, perseverance, and rational thinking and always encouraged me to pursue my interests. I dedicate this work to them.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The Internet is becoming more used for distributing content among users although IP was originally developed to interconnect pairs of users for resource sharing. This radical change has led researchers to design alternative network architectures [1–3]. One of the promising candidates is Named Data Networking (NDN) [4], which allows users to retrieve content by directly specifying the content name without referring to the address of the host holding the content. NDN carries each piece of content in a Data packet, which is identified and located with a unique content name and signed by its producer. A consumer who wishes to obtain some specific content issues an explicit request, called an Interest packet, that specifies the content name. Signatures on Data packets provide end-to-end content security: consumers can verify the integrity and provenance of content regardless of who returned it. For this reason, an Interest packet can be satisfied either by a producer or a network router caching Data packets. Such in-network caching yields several benefits in terms of the efficiency of content distribution, such as reductions in delay, bandwidth usage, and producer load.

In contrast, NDN does not provide sufficient levels of anonymity and privacy by default. The importance of anonymity and privacy has been emphasized as pervasive monitoring, censorship, and collection of personal data by ISPs, governments, and tech companies become serious and realistic concerns [5–7]. However, the following two components make anonymity and privacy difficult to provide in NDN: *human-readable content names* and *publicly verifiable signatures*. First, NDN uses hierarchically-structured human-readable content names like URL. Each content name contains the globally-routable name of its producer, called a *producer name*, as a prefix to route Interest packets toward the producer and the remaining suffix identifies specific content of the producer [8]. Such content names reveal more information regarding what content is being retrieved/published and who retrieving/publishing some specific content than the IP addresses. Second, content is inherently bound to its producer since every Data packet carries a signature, which is publicly verifiable with the producer's long-term public key. Adversaries can reveal who published some specific content from the signature.

In this thesis, the author individually designs two systems that provide particular types of anonymity and privacy by preventing information leakage from these components, augmenting anonymity and privacy in NDN to make it a serious candidate for the next-generation network architecture. The author's design is based on novel anonymity model to capture communication features specific to NDN and privacy model to complement privacy aspects not appropriately formalized in existing studies. These rigorous definitions allow the author to comprehensively analyze the levels of anonymity and privacy the proposed systems provide through theoretical and/or empirical ways.

1.1 Producer Anonymity based on Onion Routing

Background. Anonymity is currently required by journalists and whistleblowers for companies and governments, censorship circumvention systems, and privacy-sensitive applications [9–15], to conceal users' identities and online activities. Depending on who appears to be anonymous, anonymity in NDN can be classified into *consumer anonymity* and *producer anonymity*. Informally, consumer/producer anonymity mean to conceal who requests/publishes what content, respectively. In terms of unlinkability, they can be defined as *request-consumer unlinkability* and *content-producer unlinkability*, respectively. Because Interest packets do not carry any information regarding their senders, NDN naturally provides a certain level of consumer anonymity; however, this type of anonymity is weak in realistic scenarios because edge routers can still correlate Interest packets with their senders. ANDāNA [16] has been designed for NDN based on onion routing [17] to provide a level of consumer anonymity comparable to Tor [18], which is the most widely used IP-based anonymity system. ANDāNA encapsulates Interest packets with layered encryption and sends them through *circuits*, each of which consists of several *anonymizing routers*, to conceal their origins.

In contrast, naïve NDN does not provide producer anonymity owing to the information leakage from producer names and signatures and no systems to provide it have been proposed. Even worse, there exists no rigorous definition of producer anonymity. Producer anonymity on the content-oriented NDN architecture does not fit into the conventional anonymity definitions for the host-oriented IP architecture [19, 20], i.e., sender anonymity and receiver anonymity. Because request-consumer unlinkability refers to an Interest packet and its sender, consumer anonymity can be defined as a type of sender anonymity [21]. In contrast, both sender and receiver anonymity do not completely capture the notion of content-producer unlinkability because a producer neither sends nor receives any packet if an Interest packet is satisfied by a network router. This observation implies that a definition of producer anonymity should take the feature of NDN that end-to-end connections/sessions are not used into consideration.

Approach. In Chapter 3, the author rigorously defines producer anonymity and then designs a system to provide it, called ACPNDN (Anonymous Content Publishing in NDN). In contrast to sender/receiver anonymity being defined by solely focusing on hosts that sent/received a specific packet, the author defines producer anonymity by focusing on packet flow on the network. Specifically, the definition uses the notion of *indistinguishable configurations* [16, 22]. In brief, a configuration models the packet flow in a round of communication. Given the actual configuration defined according to the current network flow where a producer publishes a Data packet, the producer is said to be anonymous with respect to an adversary if, for another possible configuration where another producer publishes the Data packet, the adversary cannot correctly determine if it is observing which of the two configurations.

ACPNDN uses *pseudonyms of producers* and *rendezvous points* together with onion routing. A producer who wishes to enjoy anonymity advertises a unique pseudonym and a rendezvous point, which is an anonymizing router chosen randomly, and then builds a circuit to the rendezvous point. Consumers can retrieve content from the producer by sending Interest packets specifying the pseudonym instead of the producer name through the rendezvous point. Producer anonymity is provided because the pseudonym itself carries no information regarding the producer's identity and signatures are produced with a public/private key pair securely bound to the pseudonym. The core design of ACPNDN is based on *hidden service* [18, 23], which is implemented over Tor to provide receiver anonymity; moreover, to take full advantage of NDN, ACPNDN uses the RICE protocol [24], which is a communication protocol with different features from the original Interest-Data exchanges.

Contributions. To the best of the author's knowledge, this is the first study that rigorously defines producer anonymity taking the session-less feature of NDN into consideration and designs a realistic system to provide it. Through theoretical and empirical analysis, the author shows that ACPNDN has the following two advantages over hidden service:

First, ACPNDN reduces round trip time (RTT) required for retrieving content by building circuits over RICE's *reverse paths*, each of which consists of temporal forwarding information base (FIB) entries on a sequence of network routers. Between neighboring parties of a circuit, ACPNDN forwards Interest packets along reverse paths without using producer names, allowing producers to concealing their producer names even to their first-hop anonymizing routers. Such producer anonymity provided at the network-layer reduces the number of anonymizing routers required to provide a comparable level of anonymity to hidden service by one. Because topologically distributed anonymizing routers are chosen in realistic scenarios to minimize the probability that all the anonymizing routers of a producer's choice are compromised by an adversary, the number of anonymizing routers of a circuit considerably affects the RTT.

Second, ACPNDN offers better security against the *predecessor attack* [25], where adversaries wait until producers unfortunately choose multiple compromised anonymizing routers as members of their circuits. This attack is a major threat against onion routing-based anonymity systems and no comprehensive countermeasures have been proposed. Hidden service mitigates this attack by employing *entry guards*, which are the first-hop anonymizing routers to conceal servers' IP addresses [26]. In ACPNDN, the first-hop network routers of producers play the role of entry guards because producers' identifiers, such as MAC addresses, are revealed only to them. In addition, ACPNDN's rendezvous points, through which consumers sends Interest packets to anonymous producers, play the role of *exit guards*, which are fixed last-hop anonymizing routers like entry guards. The author shows that these changes decrease the probability of the predecessor attack succeeding.

These results shed light on the fact that it is feasible to design NDN-based anonymity systems that are more efficient and secure than existing IP-based anonymity systems by carefully leveraging a feature of NDN that a certain level of anonymity is naturally provided at the network-layer although content names and signatures in NDN reveal considerable amount of private information to adversaries. In conjunction with consumer anonymity provided by ANDāNA, ACPNDN gives a fundamental solution to provide anonymity on NDN network.

1.2 Location Privacy based on *K*-anonymity

Background. In terms of privacy, the author particularly focuses on *location privacy* of consumers. Technically, location privacy aims to conceal geographical information of current locations of users or locations of their interest (LOIs) sent for enjoying some location-based services (LBSes). There is increasing demand in collecting and analyzing such location information to improve users' online activities or develop new LBSes, such as location-based games and location-based social networking services [27]; however, the lack of location privacy poses serious attacks including user profiling (e.g., revealing private information of users, such as their homes, health conditions, lifestyles, and religion), physical harassment (e.g., physical assault, arrest, and stalking), and denial-of-service (e.g., denial of access to services from some regions). In addition, it has been pointed out that adversaries can re-identify individuals from their location information by combining with other public information even if their identities are concealed [28, 29].

One might wish to develop a one-size-fits-all solution for protecting location privacy in all possible scenarios; however, as discussed in [30], there seems to be no such solution because the definitions of location privacy vary depending on details of location information to be concealed. In addition, there are generally trade-offs between location privacy and quality-of-services, such as computation/communication costs and utility of the services. For these reasons, the research community has focused on seeking the most appropriate solution for each scenario by leveraging various techniques, such as anonymization and obfuscation of location information [31–33].

In this thesis, the author particularly focuses on an Internet-of-Things (IoT) scenario where consumers retrieve content produced by IoT devices installed at their LOIs. Such content is referred to as *location-related content*, hereinafter. As concrete NDN-based IoT platforms to efficiently retrieve location-related content, a keyword-based ICN-IoT platform [34] and name-based geographical forwarding [35, 36] have been proposed. An important feature of these platforms is that a consumer retrieves location-related content directly from their LOI by sending an Interest packet specifying the unique location name assigned to the LOI as the name prefix (i.e., producer name) and the type of desired content as the suffix. Different from producer anonymity that aims to conceal which producer is publishing some specific content, location privacy aims to conceal which producer (i.e., location) is being retrieved content by a specific consumer. For protecting location privacy of consumers, the names of consumers' LOIs and signatures produced by IoT devices carried in Interest and Data packets must be concealed from network adversaries in an end-to-end way.

Approach. In Chapter 4, the author designs a system to protect location privacy on the NDN-based IoT platforms, called PLCR (Private Location-related Content Retrieval). To prevent adversaries from identifying LOIs of consumers, PLCR leverages the notion of *k-anonymity of location* [31, 37–39]. Existing studies leverages k-anonymity of location offered by an honest *anonymizer* to protect location privacy against an LBS provider as follows: a consumer establishes an encrypted session with an anonymizer and sends the identifier of their LOI to it. The anonymizer then generates a *location anonymity set*, which comprises the LOI and other $k - 1$ dummy locations, and retrieves k pieces of content corresponding to the k locations from the LBS provider to conceal the exact

LOI. Finally, the anonymizer only sends back the piece of content corresponding to the LOI to the consumer.

This approach can be applied to the NDN-based IoT platforms: the names of LOIs and signatures by IoT devices can be concealed with encryption between consumers and the anonymizer and with k-anonymity between the anonymizer and IoT devices. However, the following two important problems remain unsolved:

Problem 1: Requirements for each location anonymity set to provide sufficiently strong k-anonymity of location have not been formalized.

Problem 2: The adversary model in the existing studies are unrealistically weak and thus the anonymizer-based system must be refined so that location privacy is protected under a more realistic adversary model.

First, a part of the existing studies proposed several requirements for each location anonymity set [37, 38, 40]; however, an LOI or its geographical information, such as the rough range in which the LOI is included, can be revealed even if the requirements are satisfied. This is because the requirements have been defined in ad-hoc manners to prevent specific attacks rather than being defined following rigorous theory of privacy. To solve **problem 1**, the author rigorously defines two requirements to achieve k-anonymity of location according to the well-established theory of data privacy: entropy and t-closeness [41]. These requirements guarantee that adversaries can infer the LOI from a location anonymity set with a probability sufficiently close to $1/k$ and the geographical information of the LOI gained by adversaries is minimized, respectively.

Second, the anonymizer in the existing studies is a single point of attack: if an adversary compromises it, the location privacy of all consumers is broken. To solve **problem 2**, the author designs PLCR under the condition that the anonymizer is a semi-honest adversary to break location privacy. The term "semi-honest" means that the anonymizer follows prescribed protocol but attempts to gain more information than allowed from the protocol. In the course of designing PLCR, the following three sub-problems must be solved:

Problem 2-1: Increase in the communication overhead must be suppressed.

Problem 2-2: Some mechanism to allow the semi-honest anonymizer to derive the popularities of locations, which are required to generate location anonymity sets, must be incorporated into the system.

Problem 2-3: Some mechanism to protect location privacy for continual content requests of a consumer must be incorporated into the system.

First, a straightforward way to realize a semi-honest anonymizer is to make the anonymizer return all the k pieces of content corresponding to a location anonymity set. This approach can conceal LOIs from the anonymizer; however, the communication overhead increases proportionately to the value of k. To circumvent this sub-problem, PLCR leverages a private information retrieval (PIR) protocol [42, 43]. Each consumer only retrieves the piece of content corresponding to their LOIs while hiding the LOIs from an anonymizer by leveraging a homomorphic encryption scheme.

Second, an honest anonymizer can derive the popularity of a location from the number of requests that specify the location as the LOI. In contrast, a semi-honest anonymizer cannot derive it since LOIs are concealed. To circumvent this sub-problem, PLCR leverages a locally differentially private (LDP) protocol [44]. Consumers periodically upload statistics on their past LOIs to a semi-honest anonymizer while hiding the exact LOIs by adding carefully controlled random noises to the statistics. The semi-honest anonymizer then estimates the popularities of locations by aggregating the statistics.

Third, the semi-honest anonymizer can infer a current LOI from a sequence of location anonymity sets used in continuous content requests by a consumer since the requirements formalized to solve **problem 1** only focuses on a single location anonymity set. To prevent such an attack, PLCR leverages the feature of NDN that consumer anonymity is naturally provided. Because PLCR allows consumers to anonymously communicate with the anonymizer, the anonymizer cannot correlate multiple content requests of consumers with each other.

Contributions. From the perspective of privacy models, the author refines the existing requirements for each location anonymity set by complementing privacy aspects not appropriately formalized. According to the requirements, the author formalizes the design rationale for location anonymity set generation algorithm as an optimization problem and then presents a concrete heuristic algorithm based on the Mondrian algorithm [45], which is a well-studied greedy algorithm for k-anonymization of databases. Through empirical analysis under a realistic IoT scenario, the author shows that the resulting location anonymity sets provide sufficiently strong k-anonymity of location.

From the perspective of system design, the author shows that a semi-honest anonymizer can be realized with reasonable communication overhead by leveraging PIR, LDP, and consumer anonymity provided on the network-layer. A disadvantage of using PIR is the additional computation overhead for encryption, multiplication, and decryption operations regarding the homomorphic encryption scheme. The author evaluates the communication and computation costs of PLCR and uncovers that it is especially preferable for consumer who have reasonable computation capabilities and prefer large values of k. PLCR itself is designed for the NDN-based IoT platforms; however, the requirements for each location anonymity set and deign rationale to realize a semi-honest anonymizer are independent of the underlying network architecture and thus can be used for protecting location privacy against LBSes on the current IP network.

Chapter 2

Related Work

2.1 Anonymity

This section summarizes existing studies regarding anonymity systems for IP and NDN.

2.1.1 Onion Routing

The design of onion routing is derived from Chaum's Mix-Net [46], which aims to conceal the senders of messages. Mix-Net encapsulates messages with layered public-key encryption and relays them through a sequence of nodes called anonymizing routers. Each anonymizing router decrypts, delays, re-orders messages before sending them to the next party specified by the sender. Subsequent onion routing-based anonymity systems are categorized into high-latency and low-latency systems.

High latency onion routing

The high-latency systems including Babel [47], Mix-master [48], and Mixminion [49] aim to provide anonymity against a global adversary, which can observe all the input and output messages of all anonymizing routers. To conceal the correlation between inputs and outputs of each anonymizing router, the high-latency systems inject decoy messages and random delays; for example, Babel relays messages in random order after batching at least N input messages. A problem of this approach is that anonymizing routers cannot relay messages within a reasonable time if input messages are scarce. To solve this problem, Babel divides time into equal periods of length T and injects decoy messages so that every anonymizing router sends N messages in each period. The high-latency systems are suitable for delay-tolerant applications like e-mail and file sharing.

Low latency onion routing: Tor, hidden service, and ANDāNA

In contrast, the low-latency systems aim to provide anonymity against a non-global adversary. To minimize extra communication overhead, each anonymizing router relays messages as fast as possible without using decoy messages or random delays. Therefore, the low-latency systems are suitable for many applications including delay-sensitive scenarios like web browsing and voice/video chatting. ACPNDN belongs to this category. The author describes Tor [18] and hidden service [18, 23] as the representative low-latency systems to provide sender and receiver anonymity on the IP network, respectively. For simplicity, the author focuses on a scenario where a client sends a content request to a server in the context of IP communication, hereinafter. According to the definition in [19], the author defines sender/receiver anonymity as unlinkability of a plaintext message and its sender/receiver, respectively.

In Tor, a client first builds a circuit by incrementally exchanging secret keys with several anonymizing routers. The client issues a message encapsulated in multiple layers of secret key encryption through the circuit. Each anonymizing router decrypts the top layer and forwards it to the next anonymizing router or the intended server. Because each message is forwarded through distributed anonymizing routers while altering its bit pattern by decryption, its origin is mixed with other possible senders from the perspective of adversaries and thus sender anonymity is achieved. Although each client periodically changes the circuit, the first-hop anonymizing router hiding the client's IP address is used repeatedly for a longer period of time. Such fixed first-hop anonymizing routers are called entry guards. Entry guards are introduced to mitigate the predecessor attack [25, 26], a type of traffic analysis attack.

Hidden service is currently deployed over Tor to allow servers to conceal their identities from other parties including clients. In the following description, the author assumes that all entities communicate through circuits. A server first generates a pseudonym called an *onion address* from their public key. The server then asks several anonymizing routers to act as *introduction points*, which relay clients' connection requests to the server. If the anonymizing routers accept the requests, the server generates a *descriptor*, which contains the IP addresses of the introduction points. The descriptor is uploaded to several anonymizing routers called *descriptor directories*. A client learns the onion address in some out-of-band way, downloads the descriptor, and asks an anonymizing router to play the role of a *rendezvous point* by building a circuit that includes the rendezvous point as the last-hop anonymizing router. The client then issues a connection request through one of the introduction points. This connection request contains the IP address of the rendezvous point and the first half of keying materials, e.g., those in the Diffie-Hellman key agreement protocol. The server establishes a connection to the client through the rendezvous point. At the same time, the server sends the second half of keying materials to the client. The client and server derive a shared secret key used to encrypt and authenticate message from these key materials. At this time, the client can send messages to the server through the rendezvous point without knowing the server's IP address.

These low-latency systems are inherently vulnerable to traffic analysis attacks [25, 50–53], where an adversary attempts to violate anonymity by leveraging on meta data of packets, such as timing and volume. To solve this problem, existing studies have proposed to incorporate lightweight attack mitigation schemes into Tor and hidden service; for example, Shmatikov et al, have proposed an adaptive padding scheme, where each anonymizing router inserts dummy packets in original packet flows when it is difficult to prevent adversaries from correlating two links using inter-packet time intervals [53]. These schemes can be leveraged in ACPNDN almost without modification; however, ACPNDN does not explicitly employ them, owing to their high communication overhead.

Next, the author describes ANDāNA, which is an initial attempt to adapt Tor to NDN to provide consumer anonymity [16]. Similar to sender anonymity in IP, consumer anonymity can be defined as request-consumer unlinkability, i.e., unlinkability of a plaintext Interest packet and a consumer who sends it. The content-oriented design of NDN is compatible with consumer anonymity. Consumer anonymity is naturally achieved against adversaries on core networks because each Interest packet carries information about which Data packet is being requested but not about who is requesting it. However, this kind of consumer anonymity is insufficient against adversaries on edge networks because they can directly observe who sends a specific Interest packet.

To solve this problem, ANDāNA [16] has been designed. ANDāNA has the advantage that it achieves a level of anonymity comparable to that of Tor with one fewer anonymizing router. The term "level of anonymity" means the number of anonymizing and network routers an adversary must compromise to break anonymity by tracking packets throughout a circuit; for example, in the case that three anonymizing routers are included in a sender's circuit in Tor, the level of anonymity is three. This is because an adversary can learn the sender's IP address by compromising the first-hop anonymizing router and track packets from the sender throughout the circuit by compromising the rest of the anonymizing routers. The above advantage is owing to the fact that only the first-hop network routers of consumers can learn their identities in ANDāNA, whereas the first-hop anonymizing routers can learn senders' IP addresses in Tor. Therefore, the adversary must compromise the first-hop network routers in addition to the anonymizing routers in circuits to break consumer anonymity.

Note that producer anonymity in NDN is briefly mentioned in [16]; however, the author presents a stronger and more rigorous definition of producer anonymity in this thesis. Intuitively, the definition of ANDāNA considers producer anonymity only against adversaries on anonymizing routers and network routers, whereas the author's definition also considers producer anonymity against consumers.

2.1.2 Anonymizer

Anonymizer-based systems provides anonymity by relaying messages through a single node called an anonymizer rather than a sequence of anonymizing routers. The role of an anonymizer is very similar to an anonymizing router in the low-latency systems: concealing the IP addresses of message senders and altering the bit-patterns of the messages. Thus, they can be regarded as a variant of the onion routing-based systems. For example, anonymizer.com provided anonymizer service as a proxy server that accesses servers on behalf of users [54]. Several anonymizer-based systems have also been proposed for NDN. Tourani et al. have proposed a lightweight mechanism to conceal content names that leverages Huffman coding on an anonymizer [55] rather than secret key encryption schemes. Kurihara et al. have proposed a censorship evasion scheme by leveraging an anonymizer [56]. The above two systems assume that anonymizers are honest.

2.1.3 Probabilistic Relaying

Crowds is a P2P-based anonymity system in IP without message encapsulation [57]. In Crowds, messages are sent around peers to conceal their origins. Each peer probabilistically decides whether to forward received messages to the intended destinations or another peer. This process makes it difficult for adversaries far away from the origin of a message to trace back the message to the origin. Inspired by Crowds, CRISP is proposed for NDN to achieve consumer anonymity [21]. In CRISP, instead of peers, each network router probabilistically determines whether to forward an Interest packet toward the intended producer or another cooperative network router. These systems have the same drawback that anonymity is broken if the first-hop entity of the sender, i.e., the neighboring peer or network router, is compromised by adversaries because such adversaries can immediately learn who initiates a particular message.

2.1.4 Dining Cryptographer Networks

Dining cryptographer networks (DC-nets) is based on the dining cryptographer problem [58], which studies a secure multi-party computation regarding the XOR operation. The problem setting is as follows: Three cryptographers are sitting down at a restaurant and the waiter informs them that the meal has been paid for by someone, either one of the cryptographers or the National Security Agency (NSA). The cryptographers can resolve their uncertainty by executing the following protocol. First, every pair of cryptographers flip a coin so that only the two of them can see the outcome. The outcome is treated as a bit. Second, each of the cryptographers announces the XOR of the two bits if they did not pay for the meal and the opposite of the XOR otherwise. Finally, the cryptographers compute the XOR of all the results. If the result equals 0, it indicates that NSA paid. Otherwise, one of the cryptographers paid but the payer remains anonymous. The above protocol can be regarded as a perfectly anonymous disclosure of a bit of information. DC-nets realizes anonymous communication by repeating the above protocol. One of the drawbacks of DC-nets is that, for a group of size N, N random bits are required to send 1 bit.

2.1.5 Group Signature

Group signatures [59, 60] are signing schemes that allow producers to sign content without being identified from the producers in the same group. For privacy-preserving signature verification, Sanjeev et al. have proposed an attribute-based signature scheme, a type of group signature scheme, suitable for NDN, called NDN-ABS (NDN Attribute-Based Signature) [61]. With NDN-ABS, consumers cannot identify a single producer among a set of producers with the same attribute from a signature. However, NDN-ABS cannot provide producer anonymity against adversaries eavesdropping packets on the network because it focuses only on the information leakage from signatures and thus producers can be identified by the producer names and packet routes. In addition, the attribute authority, which managed producers' secret keys used to generate signatures, is a single point of failure in terms of anonymity.

2.2 Location Privacy

This section summarizes existing studies on database privacy protection techniques and on location privacy protection techniques derived from them. Note that the system models assumed in the existing studies on location privacy protection are different from this thesis. Most existing studies assume that a centralized server called a LBS provider holds the content corresponding to all locations. Such a centralized model has an advantage that it can offer flexible services. Specifically, an LBS provider can answer the following two types of queries: range queries and nearest neighbor queries. A range query requests content regarding locations contained in a specified region at once (e.g., "send me a list of convenience stores within 1 km from here.”). A nearest neighbor query requests content regarding the closest target from a specified location (e.g., “send me the address of the convenience store closest to the specified location.”). In contrast, the author considers a different system model where consumers wish to retrieve content held by IoT devices installed at their exact LOIs. In such a decentralized model, the above two types of queries cannot be used; however, consumers can retrieve the latest content of each location without delays caused by passing through a centralized LBS server.

2.2.1 K-anonymity

The author describes k-anonymity and t-closeness used for database privacy in detail. Suppose a database comprises entries describing several user attributes. Attributes whose values must not be correlated with their users are called sensitive attributes (e.g., medical condition), and other attributes are called non-sensitive attributes (e.g., zip code and age). To prevent adversaries from correlating a specific user with their associated sensitive attribute, uniquely identifiable information of users (e.g., name and address) is excluded from all entries in advance in most cases. However, this is insufficient because values of non-sensitive attributes can be exploited to identify a user by combining them with other information. Such a non-sensitive attribute is called a quasi-identifier.

K-anonymity was originally proposed to prevent this attack [62]. Specifically, a database is said to be k-anonymized if, for every entry, there exist at least $k-1$ other entries that have the same values for all quasi-identifiers. Each set of entries that satisfies this condition is called an anonymity set. If a database is k-anonymized, it is difficult for adversaries to identify a specific user’s value of a sensitive attribute even when they know in which anonymity set the user is included. However, privacy leakage can still happen because k-anonymization does not consider the distribution of values of a sensitive attribute in each anonymity set. For example, a user’s value of a sensitive attribute is revealed if all entries in the anonymity set have the same value.

T-closeness was proposed to limit information that adversaries can obtain by observing the values of a sensitive attribute in an anonymity set [41]. Specifically, t-closeness requires that the difference between the distribution of values of a sensitive attribute in each anonymity set and that in the entire database is no more than a threshold t. The generation of anonymity sets satisfying t-closeness ensures that values of the sensitive attribute in each anonymity set are sufficiently distributed throughout the domain.

Several studies leverages k-anonymity to off location privacy. The first study among them proposed a spatial cloaking technique using a trusted anonymizer [31]. In this technique, a consumer first notifies their LOI to an anonymizer. The anonymizer generates a location anonymity set containing the LOI and $k - 1$ dummy locations around it and then sends the location anonymity set to an untrusted LBS provider to ensure that the LBS provider cannot identify the LOI from the k locations; however, several problems remain unsolved.

The first problem is in terms of the location anonymity set. Niu et al. claimed that the entropy of popularities of k locations in a location anonymity set should be maximized to prevent adversaries from probabilistically inferring an LOI [40, 63]. In addition, several studies claimed that location anonymity sets must not have intersections (i.e., must not include the same locations) to prevent adversaries from identifying dummy locations [37, 38]. These requirements have been proposed independently, and no studies have considered both of them. In this thesis, the author shows that both of the requirements must be satisfied simultaneously and express them collectively as a more rigorous expression.

Moreover, adversaries can learn the rough range in which an LOI is included because the location anonymity set generation algorithm in [31] simply chooses $k - 1$ locations adjacent to an LOI. To prevent this information from leaking, (k,s)-privacy states that the minimum bounding rectangle (MBR) of k locations should be greater than some constant value s [39, 64]. Similarly, Hwang et al. claimed that MBR of k locations should contain l different road segments [65]. Niu et al. claimed that $k - 1$ locations should be chosen so that the total sum/product of the distances between k locations is maximized [40, 63]. These studies require that location anonymity sets are generated by choosing dummy locations whose distances are sufficiently large; however, this approach is sometimes insufficient to minimize the geographical information leakage, as discussed in Section 4.1.3. In contrast, the author shows that the geographical information of an LOI can be minimized by making the difference in the geographical distribution of each location anonymity set with respect to that of all locations sufficiently small based on the notion of t-closeness.

The second problem is in terms of the adversary model. Because the anonymizer must be trusted by all consumers, the anonymizer is a single point of attack. Similar to this thesis, Wang et al. aimed to design a semi-honest anonymizer [66]; however, location privacy could not be protected because they focused solely on preventing the anonymizer from linking a consumer's continual requests with each other. Similarly, other studies aimed to make the anonymizer unnecessary by enabling consumers to generate their location anonymity sets on their own [40, 63]; however, they assumed a weak adversary who leverages only the popularities of locations and focused on a snapshot request by a consumer, and thus, could not protect location privacy for continual requests. In this thesis, the author designs a semi-honest anonymizer by leveraging a PIR-based scheme and the feature of NDN that consumer anonymity is naturally provided.

2.2.2 Differential Privacy

Differential privacy was originally proposed to preserve the privacy of each entry in a database, while enabling users to obtain statistics regarding the overall entries [67]. To this end, differential privacy adds carefully controlled random noise to statistics sent to users. An advantage of differential privacy is that it protects privacy regardless of the auxiliary information of adversaries.

Several studies leveraged differential privacy to protect location privacy. For example, Andres et al. proposed to add noises that follows a Laplacian-based distribution to the coordinates of LOIs to conceal their exact coordinates [32]. This technique is useful for range queries and nearest neighbor queries because the returned content does not change much even if such noises are added; however, it cannot be applied to the NDN-based IoT platforms since consumers aim to retrieve content held by IoT devices at their exact LOIs, such as photographs of specific locations. In addition, a crucial problem is that the level of location privacy is degraded for continual requests of consumers, as pointed out in [32].

2.2.3 Private Information Retrieval (PIR)

Suppose a user wishes to retrieve the i-th item of a database with $u \in \mathbb{N}$ items without revealing i. The simplest way is to retrieve all items from the database; however, the amount of data they exchange increases linearly with u. PIR was proposed to enable users to accomplish this goal with smaller communication costs by leveraging cryptographic techniques like homomorphic encryption schemes [42,43].

Several previous studies proposed PIR-based techniques to protect location privacy, especially assuming scenarios where range queries and nearest neighbor queries are used [33,68]. However, they cannot be applied to the NDN-based IoT platforms because they assume the existence of a centralized LBS provider. In this thesis, the author leverages PIR to realize location privacy protection against a semi-honest anonymizer. Specifically, PLCR provides k-anonymity of location against an anonymizer by using a PIR protocol for retrieving only the piece of content corresponding to an LOI from the pieces of content collected from all the k locations in a location anonymity set.

Chapter 3

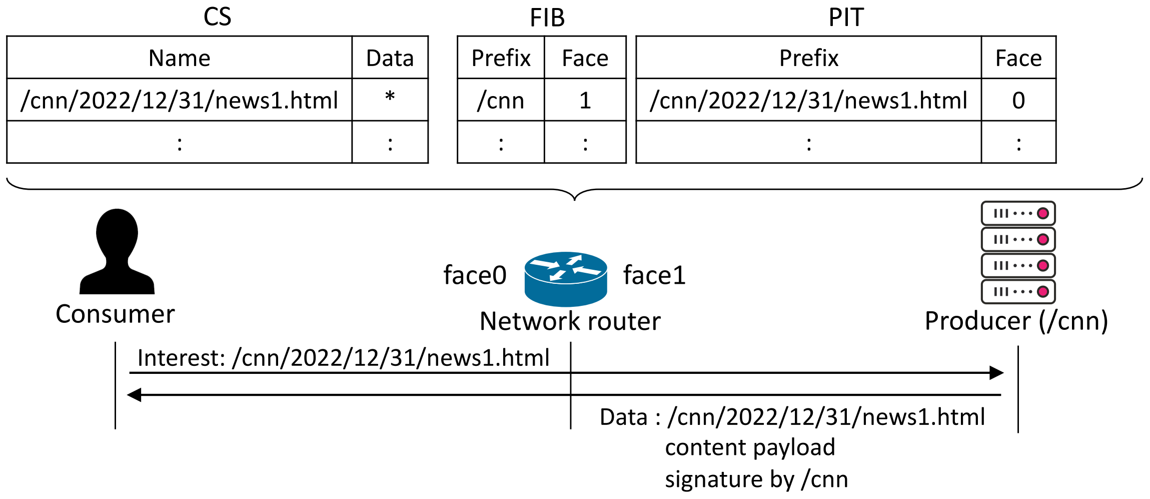
Producer Anonymity based on Onion Routing in NDN

This chapter designs ACPNDN as the first NDN-based system to provide producer anonymity under a realistic adversary model where adversaries can eavesdrop packets on arbitrary points on the network. The rest of the chapter is organized as follows: Section 3.1 describes the overview of the NDN architecture and RICE protocol leveraged by ACPNDN. Section 3.2 rigorously defines producer anonymity. Section 3.3 describes the design of ACPNDN. Section 3.4 analyzes the level of anonymity ACPNDN provides against the following two adversaries: an adversary who just observes bit patterns of packets passing through compromised entities and another adversary who also observes other sources of information, such as timing and volume of packets. Section 3.5 evaluates the performance of ACPNDN. Section 3.6 concludes this chapter.

3.1 Preliminaries

3.1.1 NDN Overview

As illustrated in Figure 3.1, the basic communication in NDN follows a *pull model*, where a consumer retrieves a piece of content by specifying the content name rather than the address of a node publishing the piece. Content names are typically hierarchically-structured human-readable names like URL. In general, a content name contains the name of a content producer, called a *producer name*, as the prefix and the remaining name components to identify a single piece of content among all the content pieces published by the producer as the suffix; for example, the first CNN news content for December



CS

Name	Data
/cnn/2022/12/31/news1.html	*
:	:

FIB

Prefix	Face
/cnn	1
:	:

PIT

Prefix	Face
/cnn/2022/12/31/news1.html	0
:	:

Figure 3.1: Overview of NDN.

31, 2022 might be named `/cnn/2022/12/31/news1.html`. The content requests are referred to as *Interest packets* and the pieces of content are carried in *Data packets*. Each Data packet consists of a content name, a content payload, and a signature produced by a producer. An Interest packet is routed toward a producer based on the content name and any network router caching Data packets can reply with the requested content. An important feature of NDN is to maintain one-to-one flow balance of packets:

- A Data packet is sent only if the corresponding Interest packet is sent; and
- A Data packet passes through the route taken by the corresponding Interest packet in the opposite direction.

Next, the author describes how each network router deals with packets. Each network router maintains the following three components:

- Content Store (CS): cache of Data packets;
- Forwarding Information Base (FIB): routing table to route Interest packet towards producers. Each entry consists of a name prefix and the corresponding outgoing interfaces (i.e., Faces); and
- Pending Interest Table (PIT): routing table to route Data packets for not-yet-satisfied Interest packets towards consumers. Each entry consists of a name prefix and the corresponding outgoing interfaces.

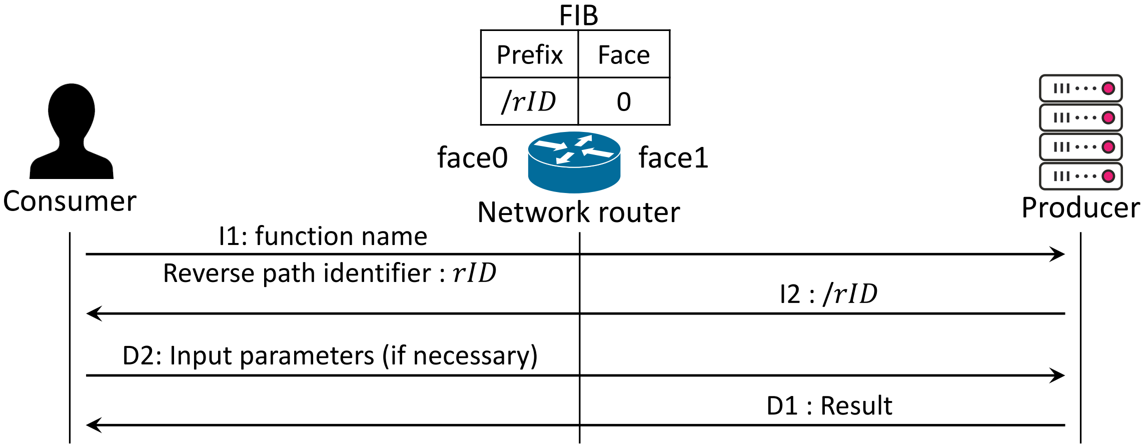


Figure 3.2: Overview of RICE.

On receiving an Interest packet, a network router first looks up its CS to check if it can reply with the corresponding Data packet. If the content name specified by the Interest packet does not match any of the Data packets, the pair of the content name and the arrival interface is recorded in a PIT entry. If the network router received multiple Interest packets with the same content name, only the first one is forwarded and the Interest packets are compiled into a single PIT entry to prevent the network router from forwarding the rest of them. After that, the unsatisfied Interest packet is forwarded according to the FIB. On receiving a Data packet, the network router forwards it to all the interfaces indicated by the corresponding PIT entry, removes the entry, and caches the Data packet. Since Data packets are forwarded toward consumers by using the sequence of PIT entries, Interest packets need not to carry any information regarding their senders and thus a level of consumer anonymity is naturally provided.

Finally, a consumer who received a Data packet verifies the signature to confirm that (1) the content payload is not tampered with (i.e., integrity); (2) the content payload was published by the intended producer (i.e., provenance); and (3) the content name is identical to that specified by the consumer (i.e., correctness). Therefore, the security of content is guaranteed in an end-to-end way even when Data packets are returned from network routers.

3.1.2 The RICE Protocol

The RICE protocol [24] is a communication protocol that has different features from the original Interest-Data exchanges in NDN. RICE was originally designed to allow consumers to delegate computation to remote parties; however, ACPNDN leverages the RICE protocol to allow producers to publish content without publicly advertising the producer names. Figure 3.2 illustrates the overview of the RICE protocol. First, a consumer issues an Interest packet (called an I1 packet) specifying the name of a function the consumer asks to execute. The I1 packet also carries a consumer-chosen *reverse path identifier*, denoted by $/rID$. On receiving the I1 packet, each intermediate network router creates an ephemeral FIB entry to forward other Interest packets (called I2 packets) specifying the reverse path identifier to the interface from which the I1 packet came. In Figure 3.2, a network router creates a FIB entry to forward I2 packets specifying the reverse path identifier as the name prefixes to face0. As a result, a sequence of FIB entries on the network routers, called a *reverse path*, is created. If a producer who has the capability to execute the function receives the I1 packet, it sends back I2 packet(s) along the reverse path to let the consumer return some input parameters for execution with the corresponding Data packet(s) (called D2 packet(s)). On receiving the D2 packet(s), the producer executes the function and returns its result with the Data packet (called a D1 packet) corresponding to the I1 packet or in another Interest/Data exchanges. An important point of the RICE protocol is that senders of I1 packets allow remote parties to send I2 packets back to them without advertising their routable names. As a result, the senders of I1 packets can *push* Data packets to the intended node.

3.2 Producer Anonymity

This section identifies several features of naïve NDN that make producer anonymity difficult to provide and then presents the adversarial model and a rigorous definition of producer anonymity.

3.2.1 Producer Anonymity in Naïve NDN

The author first describes two typical scenarios where producer anonymity is needed. In the following description, (P) and (C) denote a producer and a consumer, respectively.

- Alice (P) wishes to launch a website that provides people (C) with information about fraud by some companies or governments. She may lose her job or be punished if she publishes such information without anonymity.

- Bob (P) agrees to offer his health information, such as his age, weight, and blood pressure value, to a server for statistical surveys (C). He might wish to hide his identity from the server for his privacy.

Pfitzmann et al. defined sender anonymity and receiver anonymity as two types of anonymity on the IP architecture and they have been used as de-facto standard definitions [19]; for example, Tor provides sender anonymity against network adversaries and servers and hidden service provides receiver anonymity against network adversaries and clients. Because request-consumer unlinkability refers to a request and its sender, consumer anonymity can be defined as a type of sender anonymity [21]. Following this intuition, several systems for consumer anonymity have been designed [16, 21, 69].

In contrast, producer anonymity has not been thoroughly studied. Producer anonymity should be defined as being somewhat different from receiver anonymity. If the notion of receiver anonymity is applied to NDN, it can be defined as request-producer unlinkability, i.e., unlinkability of a plaintext Interest packet and a producer who receives it. However, the author is rather interested in content-producer unlinkability, i.e., unlinkability of a plaintext Data packet and its producer. The difference lies in the communication features of IP and NDN.

Receiver anonymity was originally defined for the host-oriented IP architecture, where two hosts communicate with each other based on end-to-end connections/sessions established between them. In contrast, the content-oriented NDN architecture does not assume such connections/sessions. In fact, every content can occasionally be returned from any intermediate entity caching it. In such a case, producers do not receive any packets, however, content-producer unlinkability can still be violated because Data packets are strongly bound to their producers by their two components, producer names and signatures.

First, a producer name is a globally routable name prefix of every content name of a producer. Each producer name plays the dual roles of the identifier and the locator of a producer simultaneously. This implies that one producer name reveals a similar amount of information to a pair of an IP address and a domain name, which is often encrypted by using cryptographic protocols including TLS [70, 71] and QUIC [72]. Therefore, each producer name carries enough information to uniquely identify the producer and to obtain more meaningful information, such as the producer's locations and affiliations [8].

Second, the signature carried in each Data packet is also regarded as the producer's identifier because it is publicly verifiable with their unique public key. These features make producer anonymity difficult to achieve, in contrast to consumer anonymity being naturally provided by the feature that Interest packets do not carry any information regarding their senders.

3.2.2 Adversarial Model

Table 3.1 summarizes the notation used throughout this chapter. $\mathbb{C}$, $\mathbb{P}$, $\mathbb{A}$, and $\mathbb{R}$ denote the sets of all consumers, producers, anonymizing routers, and network routers, respectively. Each intersection of these sets can be non-empty.

Table 3.1: Summary of notation.

Notation	Description
$\mathbb{C}$	Set of all consumers
$\mathbb{P}$	Set of all producers
$\mathbb{A}$	Set of all anonymizing routers
$\mathbb{R}$	Set of all network routers
$\mathbb{D}$	Set of all valid Data packets
$\mathcal{A}$	Adversary
κ	Security parameter
CF	Configuration
$\perp$	Special symbol meaning no circuit
(pk_{id}, sk_{id})	Identity key pair of a producer
k_i	Secret key exchanged between a producer and the i-th anonymizing router
sID_i	Session identifier exchanged between a producer and the i-th anonymizing router
rID_i	Reverse path identifier assigned for the i-th link between anonymizing routers by a producer
H	Cryptographic hash function
$\mathsf{Cert}(pk)$	Public key certificate of public key pk
Enc_{k_i}	Secret key encryption algorithm with secret key k_i
Dec_{k_i}	Secret key decryption algorithm with secret key k_i
σ_{sk}	Signature generated with private key sk
t_{k_i}	MAC tag generated with secret key k_i
$f_{\mathbb{A}}$	Fractions of compromised anonymizing routers
$f_{\mathbb{R}}$	Fractions of compromised network routers
q	Probability that an anonymizing router becomes unavailable in a round
m	Number of rounds

The author assumes that the goal of the adversary $\mathcal{A}$ is to identify who publishes what content. Following the adversarial model in Tor and ANDāNA [16, 18], the author assumes that $\mathcal{A}$ is non-global, active, and efficient. First, the author assumes that $\mathcal{A}$ compromises only a proper subset of entities. Thus, $\mathcal{A}dv$ can be represented as a 4-tuple: $\mathcal{A} = (\mathbb{C}_{\mathcal{A}}, \mathbb{P}_{\mathcal{A}}, \mathbb{A}_{\mathcal{A}}, \mathbb{R}_{\mathcal{A}}) \subset (\mathbb{C}, \mathbb{P}, \mathbb{A}, \mathbb{R})$, where $\mathbb{C}_{\mathcal{A}}$, $\mathbb{P}_{\mathcal{A}}$, $\mathbb{A}_{\mathcal{A}}$, and $\mathbb{R}_{\mathcal{A}}$ are the sets of compromised consumers, producers, anonymizing routers, and network routers, respectively. This assumption is reasonable because these entities are assumed to be distributed throughout the networks. Second, $\mathcal{A}$ can perform any actions that the compromised entities can perform, such as eavesdropping, modification, and dropping of packets. Third, $\mathcal{A}$ runs any algorithms only in time polynomial in the security parameter $\kappa \in \mathbb{N}$. This is a fundamental assumption for almost all of modern cryptographic protocols [73].

Note that the author does not assume that $\mathcal{A}$ does not aim to block some specific content throughout the networks, such as worldwide censorship authorities. To evade such censorship, Interest and Data packets must be encrypted in an end-to-end manner in exchange for the advantage of content caching because censorship can be enforced by simply dropping Interest/Data packets which contain some censored keywords even if their origins are anonymous [74, 75]. In addition, the author does not aim to achieve consumer anonymity against $\mathcal{A}$, whereas hidden service is designed so that it provides both sender and receiver anonymity. This allows ACPNDN to leverage cached content close to consumers because circuits are not needed between consumers and rendezvous points. Moreover, as described in Section 3.4, leveraging cached content improves producer anonymity since producers do not send/receive any packets.

3.2.3 Anonymity Definition

The author presents a formal definition of producer anonymity in terms of content-producer unlinkability. Producer anonymity is defined by using the notion of indistinguishable configurations [16, 22]. In brief, a *configuration* consists of packets and network entities forwarding them and represents the packet flow in a *round* of communication. The author defines a round as a series of content publishing of producers performed without changing their circuits. For simplicity, the author assumes that each producer is requested at most one piece of content by a consumer in each round. This assumption does not affect the definition of producer anonymity because producer anonymity is broken if a producer is linked to even a piece of content.

Formally, a configuration CF is defined as a mapping which associates producers with established circuits, consumers who issue Interest packets, and the corresponding plaintext Data packets, as follows:

Definition 3.2.1 (Configuration)

$$\mathsf{CF} : \mathbb{P} \to \mathbb{A}^n \cup \{\perp\} \times \mathbb{C} \times \mathbb{D},$$

where $\perp$ is a special symbol to represent the case where content is returned to a consumer from a cache on a network router without using a circuit and $\mathbb{D}$ is the set of all the Data packets which follow the prescribed packet format.

For simplicity of notation, the author also defines the following four mappings that represent elements in a configuration CF; $\mathsf{CF}_{\mathbb{A}} : \mathbb{P} \to \mathbb{A}^n \cup \{\perp\}$ (selections of n anonymizing routers in circuits), $\mathsf{CF}_{\mathbb{A}_i} : \mathbb{P} \to \mathbb{A}$ (selections of i-th anonymizing routers in circuits), $\mathsf{CF}_{\mathbb{C}} : \mathbb{P} \to \mathbb{C}$

(associations between producers and consumers), and $\mathsf{CF}_{\mathbb{D}} : \mathbb{P} \longrightarrow \mathbb{D}$ (selections of Data packets to publish). For example, if a producer $p \in \mathbb{P}$ publishes a Data packet *dat* $\in \mathbb{D}$ along a circuit consisting of n anonymizing routers $\{a_1, \cdots, a_n\} \in \mathbb{A}^n$ to a consumer $c \in \mathbb{C}$ in a configuration CF, then $\mathsf{CF}(p) = \{a_1, \cdots, a_n, c, dat\}$, $\mathsf{CF}_{\mathbb{A}}(p) = \{a_1, \cdots, a_n\}$, $\mathsf{CF}_{\mathbb{A}_i}(p) = a_i$, $\mathsf{CF}_{\mathbb{C}}(p) = c$, and $\mathsf{CF}_{\mathbb{D}}(p) = dat$. For another example, if c receives *dat* of p from a cache on a network router in CF, then $\mathsf{CF}(p) = \{\perp, c, dat\}$. In this case, p does not send/receive any packets.

Because $\mathcal{A}$ can eavesdrop packets only at a portion of entities and Data packets are encrypted throughout circuits, for a configuration CF, there can exist another possible configuration CF' which yields packet flow that seem to be consistent with those yielded in CF from the viewpoint of $\mathcal{A}$. In this case, $\mathcal{A}$ cannot differentiate CF from CF', i.e., CF and CF' are indistinguishable with respect to $\mathcal{A}$. To formalize this notion, let $\mathcal{A}(1^\kappa, \widehat{\mathsf{CF}})$ denote any probabilistic algorithm run by $\mathcal{A}$ in time polynomial in the security parameter κ such that, given $\widehat{\mathsf{CF}}$ chosen uniformly at random from a known set $\{\mathsf{CF}, \mathsf{CF}'\}$, it outputs 1 if it deduces that $\widehat{\mathsf{CF}}$ corresponds to CF. From this definition, $\Pr[\mathcal{A}(1^\kappa, \mathsf{CF}) = 1]$ and $\Pr[\mathcal{A}(1^\kappa, \mathsf{CF}') = 1]$ represent the probability that $\mathcal{A}$ correctly identifies inputted CF as CF and the probability that $\mathcal{A}$ incorrectly identifies inputted CF' as CF, respectively. By using $\mathcal{A}(1^\kappa, \widehat{\mathsf{CF}})$, indistinguishable configurations can be defined as follows:

Definition 3.2.2 (Indistinguishable configurations) *Two configurations* CF *and* CF' *are said to be indistinguishable with respect to* $\mathcal{A}$*, denoted as* $\mathsf{CF} \equiv_{\mathcal{A}} \mathsf{CF}'$*, if for* $\mathcal{A}$ *there exists a negligible function* $\epsilon(\cdot)$*, such that*

$$|\Pr[\mathcal{A}(1^\kappa, \mathsf{CF}) = 1] - \Pr[\mathcal{A}(1^\kappa, \mathsf{CF}') = 1]| \leq \epsilon(\kappa),$$

for the security parameter κ*.*

From the definition, the left side of the inequality in definition 3.2.2 represents the probability that $\mathcal{A}$ can distinguish CF from CF'. A function is said to be negligible if it is asymptotically smaller than an inverse function of any positive polynomial. Therefore, $\mathsf{CF} \equiv_{\mathcal{A}} \mathsf{CF}'$ implies that $\mathcal{A}$ can correctly differentiate CF and CF' only with negligible probability.

Content-producer unlinkability is achieved with respect to $\mathcal{A}$ if $\mathcal{A}$ can determine neither which content a producer $p \in \mathbb{P}$ is providing nor whether p or another producer $p' \in \mathbb{P}$ $(p' \neq p)$ is publishing specific content. This notion can be formalized by using indistinguishable configurations as follows: given the actual configuration CF (i.e., the configuration which reflects the actual network activities $\mathcal{A}$ is observing) in which p publishes a Data packet *dat*, content-producer unlinkability is achieved if there exist another imaginary but possible configuration CF' in which p' publishes *dat* and p publishes another Data packet, and $\mathcal{A}$ cannot determine whether it is observing either CF or CF'. This implies that content publishing of p and p' causes only a negligible difference in $\mathcal{A}$'s observation. Producer anonymity in terms of content-producer unlinkability can be defined as follows:

Definition 3.2.3 (Producer anonymity) $p \in (\mathbb{P} \setminus \mathbb{P}_{\mathcal{A}})$ *has producer anonymity in configuration* CF *with respect to* $\mathcal{A}$ *if* $\exists \mathsf{CF}' \equiv_{\mathcal{A}} \mathsf{CF}$ *such that* $\exists p' \in (\mathbb{P} \setminus \mathbb{P}_{\mathcal{A}})$, $\mathsf{CF}'_{\mathbb{D}}(p') = \mathsf{CF}_{\mathbb{D}}(p) \neq \mathsf{CF}_{\mathbb{D}}(p') = \mathsf{CF}'_{\mathbb{D}}(p)$ *and* $p' \neq p$.

From the perspective of an anonymity set, which is generally defined as the set of all possible subjects that might cause an action [19], the anonymity set with respect to p's content publishing consists of p and all the producers who satisfy the requirements for p' in the definition 3.2.3. As the number of producers in the anonymity set increases, p is hidden in the larger crowd and thus the anonymity degree increases.

3.3 Anonymous Content Publishing in NDN

This section first provides the overview of ACPNDN and then describes the overall communication procedure in detail.

3.3.1 System Model

One of the key constraints to provide producer anonymity is that producers cannot initiate content publishing without receiving Interest packets from consumers in NDN, as described in Section 3.1. Taking this constraint into account, ACPNDN uses pseudonyms of producers, called *onion names*, and *rendezvous points*. Onion names are used to distinguish services provided by anonymous producers; however, consumers cannot send Interest packets toward producers only by using onion names because onion names are designed so that their producers cannot be identified and located as described in Section 3.3.2. Therefore, rendezvous points accept Interest packets specifying onion names under their own globally routable names and relay them to anonymous producers.

The overview of ACPNDN is illustrated in Figure 3.3. The author assumes that every anonymizing router advertises its routable name and its public key certificate via directory nodes like Tor and hidden service.

- A producer who wishes to anonymously publish content generates a long-term public/private key pair and derives an onion name from the public key.

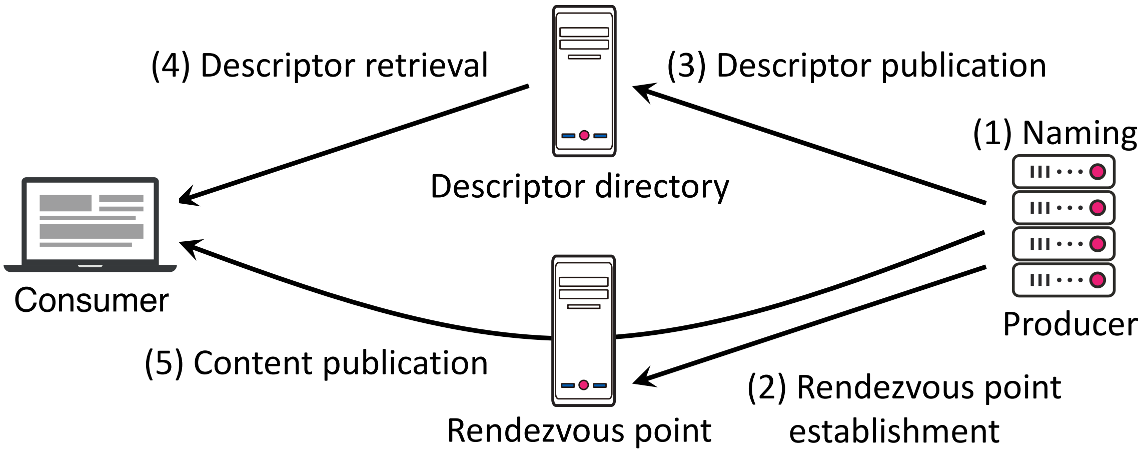


Figure 3.3: Overview of ACPNDN.

- The producer asks an anonymizing router to act as a rendezvous point through a circuit. If the anonymizing router accepts it, the producer waits for content requests while maintaining the circuit. Circuits are periodically re-built by the producer like Tor and hidden service; however, the rendezvous point is used for a longer period of time until it becomes unavailable.

- The producer generates a descriptor contains the producer's public key certificate, the rendezvous point's routable name, and the rendezvous point's public key certificate. The producer then uploads the descriptor to some of the *descriptor directories*, which are anonymizing routers for publishing descriptors, through another circuit. The name of the descriptor is derived from the corresponding onion name.

- A consumer who learns the onion name in some out-of-band way (e.g., web pages summarizing onion names and their services) downloads the descriptor from one of the responsible descriptor directories by specifying the descriptor name.

- To obtain content, the consumer issues a request toward the rendezvous points. The corresponding content is returned by a network router between the consumer and the rendezvous point if it has cached the content. Otherwise, the rendezvous point relays the request to the producer through the circuit.

Throughout this procedure, the producer can publish content without revealing anything more than the onion name.

Note that the consumer builds circuits neither to the descriptor directory nor the rendezvous point since this chapter focuses solely on producer anonymity. In addition, hidden service uses introduction points to enable senders to send connection requests to receivers and exchange secret keys with them, as described in Section 2.1.1. This additional communication phase is required to establish connections and to encrypt packets in an end-to-end manner with the secret keys to evade censorship enforced throughout the networks; however, ACPNDN does not use introduction points because the author does not assume such worldwide censorship, as described in Section 3.2.2.

3.3.2 Naming

In ACPNDN, an onion name is used as a prefix of every content name of a producer instead of their producer name. There are several requirements for onion names that are different from those for producer names. It must be ensured that every onion name is non-routable, non-human-readable, and unique and securely bound to both its producer and their public key without relying on any authorities.

First, if an onion name is advertised as the routable name of a producer, $\mathcal{A}$ can easily correlate the onion name with the producer and thus producer anonymity cannot be provided. Second, onion names should not be human-readable to prevent information leakage from themselves. Third, in the case of producer names, uniqueness of names and bindings between a producer, their name, and their public key are established by trusted authorities, such as ICANN and CAs [76, 77]. However, ACPNDN cannot leverage such authorities to avoid any single point of failure in terms of anonymity.

Taking these requirements into account, producers generate their onion names from fresh public keys. A producer first generates a long-term public/private key pair (pk_{id},sk_{id}) called an *identity key pair*, and the corresponding self-signed public key certificate $\mathsf{Cert}(pk_{id})$ signed with sk_{id}. Note that $\mathsf{Cert}(pk_{id})$ must be generated so that it does not contain its producer's identifiers except the public key. The length of this key pair is assumed to be a function of the security parameter κ. The producer uses "onion" as the first component and the hash of pk_{id} as the second component of the onion name, respectively. By using the onion name, content names of the producer are represented as follows:

$$/\mathsf{onion}/H(pk_{id})/\langle\mathsf{suffix}\rangle,$$

where $\langle\mathsf{suffix}\rangle$ denotes the name suffixes determined by the producer, e.g., $\langle\mathsf{suffix}\rangle$ = article/xyz.html.

Such onion names satisfy the three requirements. Onion names are not routable because they are just hashes of public keys generated locally. Thus, consumers cannot send Interest packets directly to producers. For the same reason, onion names are non-human-readable and a collision of onion names occurs only with negligible probability. Finally, the bindings between a producer, their onion name, and their public key are securely established as follows: the producer is bound with the onion name and the public key because their ownership can be proved with a signature which can be generated only with the producer's private key corresponding to the public key pk_{id}, and the onion name is bound with the public key because the onion name is self-certifying, i.e., the onion name contains the hash of the public key. In addition, no authorities are required while establishing these bindings because the public key certificate $\mathsf{Cert}(pk_{id})$ and onion names are locally generated.

In terms of trust of producers, there is an inherent conflict between producer anonymity and the trust mechanism of naṽe NDN, where a consumer of a piece of content verifies its producer's certificate along the trust chain and trusts the producer if the trust chain reaches one of the consumer's trust anchors. In contrast, ACPNDN has no built-in mechanisms to provide consumers with the information they need to decide onion names to trust since self-signed certificates are used to keep private the bindings between producers, their names, and their public keys. Therefore, if necessary, trust should be established to producers' onion names instead of their identities by using reputation systems or in some ad-hoc manners [78–80]; for example, there are several websites publishing the lists of what kind of services are offered under some onion addresses in the current hidden service. The author believes that such mechanisms help consumers decide onion names whose content they retrieve and encourages anonymous producers to behave trustworthily.

Similarly, consumers cannot directly authenticate anonymous producers. Thus, consumers authenticate onion names instead. When a consumer receives a Data packet belonging to an onion name of their interest, the consumer confirms that it is certainly created by the correct owner of the onion name by verifying whether the signature is valid for the public key corresponding to the onion name. Since onion names and public key certificates reveal nothing about producers' identities, producer anonymity cannot be broken in this kind of authentication process.

3.3.3 Rendezvous Point Establishment

Next, the producer asks an anonymizing router to act as a rendezvous point by sending it the onion name and the public key certificate $\mathsf{Cert}(pk_{id})$. This anonymizing router is simply referred to as a rendezvous point, hereinafter. To prevent $\mathcal{A}$ from impersonating the producer, this request should contain the signature generated with sk_{id}, denoted by $\sigma_{sk_{id}}$. The rendezvous point accepts it only if both the onion name and $\sigma_{sk_{id}}$ are valid for pk_{id} obtained from $\mathsf{Cert}(pk_{id})$.

The important point is how to send them to the rendezvous point. It might be problematic to send them with Interest packets because they are not designed to carry much data. A straightforward way

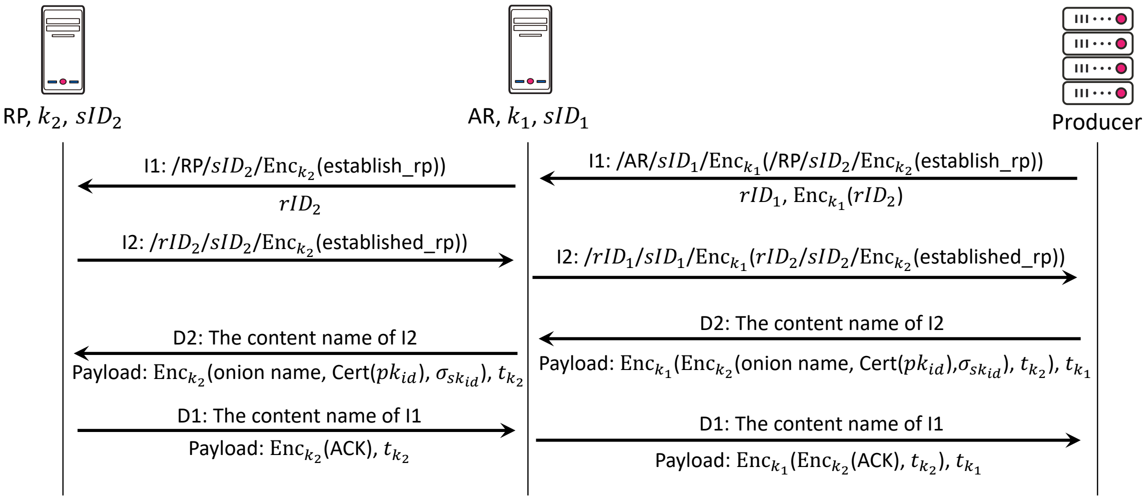

RP, k_2, sID_2
AR, k_1, sID_1
Producer
I1: /RP/sID_2/Enc$_{k_2}$(establish_rp))
rID_2
I2: /rID_2/sID_2/Enc$_{k_2}$(established_rp))
D2: The content name of I2
Payload: Enc$_{k_2}$(onion name, Cert(pk_{id}), $\sigma_{sk_{id}}$), t_{k_2}
D1: The content name of I1
Payload: Enc$_{k_2}$(ACK), t_{k_2}
I1: /AR/sID_1/Enc$_{k_1}$(/RP/sID_2/Enc$_{k_2}$(establish_rp))
rID_1, Enc$_{k_1}$(rID_2)
I2: /rID_1/sID_1/Enc$_{k_1}$(rID_2/sID_2/Enc$_{k_2}$(established_rp))
D2: The content name of I2
Payload: Enc$_{k_1}$(Enc$_{k_2}$(onion name, Cert(pk_{id}),$\sigma_{sk_{id}}$), t_{k_2}), t_{k_1}
D1: The content name of I1
Payload: Enc$_{k_1}$(Enc$_{k_2}$(ACK), t_{k_2}), t_{k_1}

Figure 3.4: Anonymous rendezvous point establishment.

to send them with Data packets is that the producer advertises their producer name to the first-hop anonymizing router to have it forward Interest packets from the rendezvous point requesting them. This is the same approach as hidden service, where receivers' IP addresses are given to the first-hop anonymizing routers as source addresses. In contrast, the producer has the first-hop anonymizing router forward such Interest packets along a reverse path used in RICE in ACPNDN. This enables the producer to send Data packets without revealing their identity even to the first-hop anonymizing router.

The author describes how the producer can establish the rendezvous point through a circuit built on reverse paths. Figure 3.4 shows the communication sequence, where a circuit includes one anonymizing router other than the rendezvous point. Let AR and RP denote the anonymizing router and the rendezvous point, and $/\mathsf{AR}$ and $/\mathsf{RP}$ denote their routable names, respectively. The author uses any CCA-secure secret key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, where Gen is a key generation algorithm which generates a secret key according to inputted security parameter κ and Enc and Dec are an encryption and a decryption algorithm with the secret key, respectively. CCA-secure encryption schemes are probabilistic and non-malleable (i.e., ciphertexts are randomized so that $\mathcal{A}$ cannot gain any partial information on the plaintexts and, given a ciphertext, $\mathcal{A}$ cannot generate a different ciphertext such that their plaintexts are somehow related [73]). The author assumes that the producer built a circuit anonymously by exchanging secret key $k_i \leftarrow \mathsf{Gen}(\kappa)$ and session identifier sID_i chosen uniformly and independently at random from $\{0, 1\}^\kappa$ with AR and RP. This can be done with standard Interest/Data packets exchanges [81].

The producer first issues an I1 packet which includes $\mathsf{establish\_rp}$ as a name component to RP to request rendezvous point establishment. The I1 packet is encapsulated in multi-layers of encryption by using Enc. The author assumes that Interest packets from the producer always pass through one network router offered by an ISP as the first-hop network router, hereinafter. AR and RP remove the top layers of the received I1 packets by using Dec with the secret keys k_1 and k_2 corresponding to the session identifiers sID_1 and sID_2 specified in the I1 packet, respectively. The I1 packet also carries reverse path identifiers rID_1 and rID_2 in each layer to create two reverse paths between the producer and AR and between AR and RP. rID_1 and rID_2 are chosen uniformly and independently at random from $\{0, 1\}^\kappa$ to prevent $\mathcal{A}$ from linking the incoming and the outgoing packet at a non-compromised anonymizing router. On the receipt of the I1 packet, RP issues an I2 packet specifying the content name $/rID_2/sID_2/\mathsf{Enc}_{k_2}(\mathsf{established\_rp})$ along the reverse path. This I2 packet notifies the producer that RP has agreed to act as the rendezvous point and is requesting the D2 packet containing the onion name, $\mathsf{Cert}(pk_{id})$, and $\sigma_{sk_{id}}$. AR encrypts the entire content name of the I2 packet with k_1 and appends rID_1 as the new name prefix to forward it along the reverse path to the producer. The D2 packet is transported by using PIT entries created by the I2 packet, while being decrypted with k_1 and k_2. The D2 packet also contains MAC tags generated with k_1 and k_2, denoted by t_{k_1} and t_{k_2}, to enable each anonymizing router to verify the origin of the D2 packet. Finally, after receiving the D2 packet, the rendezvous point returns the D1 packet corresponding the I1 packet to notify that the D2 packet has been received.

Because the I1/D2 packets do not carry their senders' identities, from the Interest/Data packet exchanges, AR and RP cannot learn their predecessors, i.e, they cannot learn the producer and AR, respectively. In contrast, the first-hop network router can learn the MAC addresses of the producer.

The rendezvous point establishment protocol requires the network routers to maintain ephemeral FIB entries, each of them contains a unique reverse path identifier, to ensure the reachability to producers. If there are so many producers who wish to enjoy anonymity, the FIB size on each network router might exceed its capability. To solve this issue, producers can leverage aggregatable reverse path identifiers by appending topological prefixes to the reverse path identifiers: $/\langle\mathsf{topological\text{-}prefix}\rangle/rID$, where $/\langle\mathsf{topological\text{-}prefix}\rangle$ denotes (maybe hierarchical) name prefixes which represent topological information of producers; however, the topological information should be carefully controlled because the use of such prefixes can degrade anonymity.

3.3.4 Descriptor Publication/Retrieval

To advertise the existence, the producer uploads the descriptor to several descriptor directories in the same way as the rendezvous point establishment phase. The descriptor is used by consumers to find the established rendezvous point corresponding to the onion name of their interests. Concretely, the descriptor is a type of content generated by the producer containing the the producer's public key certificate $\mathsf{Cert}(pk_{id})$, the routable name of the rendezvous point, its public key certificate, and the signature $\sigma_{sk_{id}}$. The author assumes that the selection of responsible descriptor directories follows previous studies on hidden service [23, 82]. In short, the descriptor directories are managed by a scheme based on a distributed hash table (DHT) and the responsible directories are determined by the content name of the descriptor (called *descriptor name*) and current timestamp. Similar to onion names, descriptor names are derived as follows:

$$/\mathsf{onion}/H(pk_{id})/\mathsf{descriptor}.$$

A consumer who learns the onion name derives the descriptor name, finds the responsible descriptor directories determined by the descriptor name, and downloads the descriptor from one of them. The consumer accepts the descriptor only if $\sigma_{sk_{id}}$ is valid for pk_{id} obtained from $\mathsf{Cert}(pk_{id})$ in the descriptor.

3.3.5 Content Publication

After uploading the descriptor, the producer waits for content requests from consumers. Because reverse paths expire after a certain amount of time has elapsed, the producer updates them by issuing I1 packets carrying nonces to RP. These I1 packets also have the role of keeping the circuit alive by sending packets periodically, similar to PADDING cells in Tor [83]. RP waits for content requests for a certain time period T determined according to the reverse path expiry time and RTT between the producer and RP. If no content request from consumers has arrived within T, RP returns a Data packet to the producer to inform that there is no request. Suppose that the expiry time of FIB entries on reverse paths is set to t_{FIB}, then $T \leq t_{FIB} - RTT$ should hold, where RTT is the estimated RTT between the producer and RP.

Figure 3.5 illustrates the flow of Interest/Data packets in the case where there is a content request from a consumer. Since the onion name is not routable, the consumer issues an Interest packet *int* requesting content through RP by appending its routable name as the content name prefix. For example, *int* carries the content name /RP/onion/encode(pk_{id})/article/xyz.html. *int* can be satisfied

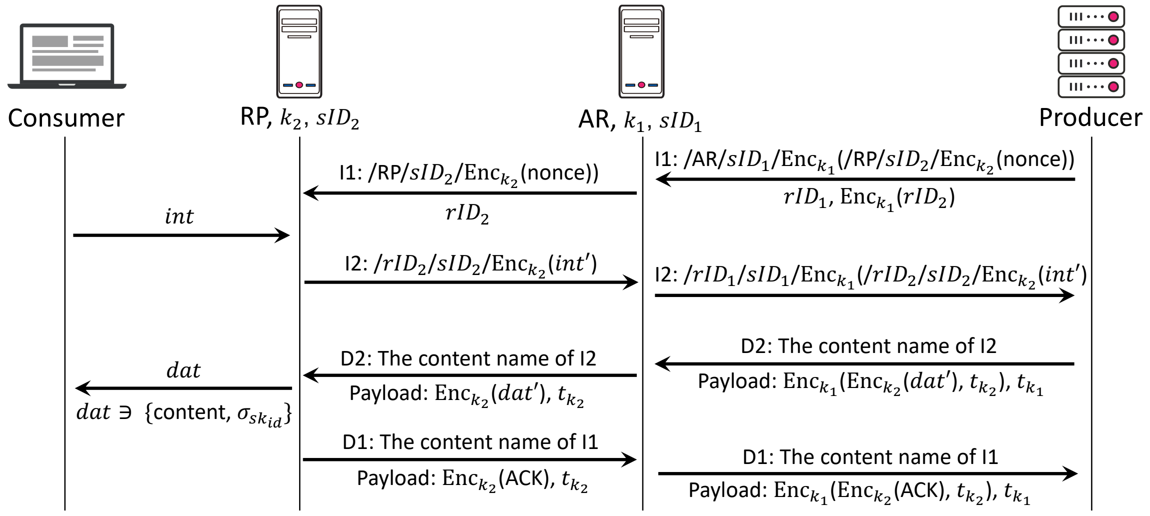

Consumer
RP, k_2, sID_2
AR, k_1, sID_1
Producer
I1: /RP/sID_2/Enc$_{k_2}$(nonce))
rID_2
int
I1: /AR/sID_1/Enc$_{k_1}$(/RP/sID_2/Enc$_{k_2}$(nonce))
rID_1, Enc$_{k_1}$(rID_2)
I2: /rID_2/sID_2/Enc$_{k_2}$(int')
I2: /rID_1/sID_1/Enc$_{k_1}$(/rID_2/sID_2/Enc$_{k_2}$(int')
D2: The content name of I2
Payload: Enc$_{k_1}$(Enc$_{k_2}$(dat'), t_{k_2}), t_{k_1}
dat
$dat \ni$ {content, $\sigma_{sk_{id}}$}
D2: The content name of I2
Payload: Enc$_{k_2}$(dat'), t_{k_2}
D1: The content name of I1
Payload: Enc$_{k_2}$(ACK), t_{k_2}
D1: The content name of I1
Payload: Enc$_{k_1}$(Enc$_{k_2}$(ACK), t_{k_2}), t_{k_1}

Figure 3.5: Anonymous content publication.

by any cache on the network routers between the consumer and RP because it is not encrypted. If *int* reaches RP without being satisfied by the caches, RP first removes the name component /RP from *int* and then forwards such a new Interest packet *int′* along the reverse path associated with the onion name specified in *int′*. On the reverse path, *int′* is treated as an I2 packet. The corresponding Data packet *dat′* containing the requested content and $\sigma_{sk_{id}}$ is returned from the producer to RP as the D2 packet by using PIT entries. Then, RP generates a Data packet *dat* which has the same content name as *int* by encapsulating *dat′* (without encryption). After sending *dat* to the consumer, RP returns the D1 packet to acknowledge the D2 packet. Suppose that the PIT entries expiry time is set to t_{PIT}, $T \leq t_{PIT} - 2RTT$ should also hold to transport the D1 packet to the producer.

The consumer verifies that *dat* has certainly been generated by the intended producer advertising the onion name by verifying $\sigma_{sk_{id}}$ in *dat* with the public key *pk* corresponding to the onion name of their interest. In addition, each anonymizing router cannot learn its predecessor in these Interest/Data packet exchanges for the same reason as the rendezvous point establishment phase.

3.4 Anonymity Analysis

This section presents an analysis of the levels of anonymity ACPNDN provides against the following two kinds of adversary: $\mathcal{A}$ that just observes bit patterns of packets passing through compromised entities and $\mathcal{A}$ that also observes other sources of information, such as timing and volume of packets. The author calls the former adversary a *weak adversary*, denoted by $\mathcal{A}^w$, and the latter adversary a *strong adversary*, denoted by $\mathcal{A}^s$, respectively ($\mathcal{A} \in \{\mathcal{A}^w, \mathcal{A}^s\}$).

3.4.1 Notation

In the following description, the author focuses on the content publication phase described in Section 3.3.5 because producer anonymity is achieved in other phases in the same way. In terms of linkability of packets, I1, I2, D1, and D2 packets traverse the same route by using PIT or reverse paths and are easily linkable by observing their content names and the reverse path identifiers. Consequently, in terms of producer anonymity, it is sufficient to focus on linkability between producers and one of these packets. In addition, I2 and D2 packets between an RP and a producer are encrypted forms of Interest and Data packets between a consumer and the PR, respectively. From these observations, the author focuses only on linkability between a producer and a Data packet.

Let $\mathcal{E}_k$ denote an operation to encrypt a Data packet once by using an encryption algorithm Enc with a secret key $k \leftarrow \mathsf{Gen}(\kappa)$ and to append a reverse path identifier and a session identifier chosen uniformly and independently at random from $\{0,1\}^\kappa$ to the content name as described in Section 3.3. A Data packet *dat* which has gone through $\mathcal{E}$ for a sequence of l secret keys $\mathcal{K}_l = (k_1, \cdots, k_l)$ (if $l = 0$, then $\mathcal{K}_l = \emptyset$) in this order is represented as follows:

$$dat_{\mathcal{K}_l} = \begin{cases} dat & (l = 0) \\ \mathcal{E}_{k_l}(\mathcal{E}_{k_{l-1}}(\cdots(\mathcal{E}_{k_1}(dat))\cdots)) & (l \geq 1) \end{cases}$$

Obviously, producer anonymity is broken if $\mathcal{A}$ can correctly correlate an outgoing Data packet at the producer, i.e., $\mathcal{E}_{k_n}(\mathcal{E}_{k_{n-1}}(\cdots(\mathcal{E}_{k_1}(dat))\cdots))$, and an outgoing Data packet at a rendezvous point, i.e., *dat*, because it implies that $\mathcal{A}$ can correlate the input and output of a circuit. In such a case, the circuit is said to be compromised by $\mathcal{A}$ in such a case.

In ACPNDN, the sender of each Data packet might be included in an anonymity set, i.e., there might exist several possible senders, from the viewpoint of $\mathcal{A}$. This is because Data packets do not carry any identifiers of their senders and the author assumes that $\mathcal{A}$ does not compromise all the entities. Note that the term “sender” does not always correspond to the producer; for example, if an anonymizing router forwards a Data packet after removing the top layer of encryption, its sender is the anonymizing router. The author defines a *sender anonymity set* of a Data packet $dat_{\mathcal{K}_l}$ with respect to $\mathcal{A}$ as follows:

Definition 3.4.1 (Sender anonymity set)

$$\mathsf{AS}_{\mathcal{A}}^{dat_{\mathcal{K}_l}} = \{e \in \mathbb{P} \cup \mathbb{A} \mid \mathsf{Pr}[\mathcal{A} \textit{ infers that } e \textit{ sent } dat_{\mathcal{K}_l} \mid \mathcal{A} \textit{ observes } dat_{\mathcal{K}_l}] > 0\}.$$

This means that the sender anonymity set of $dat_{\mathcal{K}_l}$ with respect to $\mathcal{A}$ contains all the entities which seem to have sent it with non-zero probability from the perspective of $\mathcal{A}$. When the producer sends a Data packet, $\mathcal{A}$ on the first-hop network router can identify the producer (i.e., the sender), however, the sender anonymity set will grow as it is transported toward network routers on core networks because packets from more senders can pass through them. The author assumes that the anonymity degree of all the possible senders of $dat_{\mathcal{K}_l}$ equals $|\mathsf{AS}_{\mathcal{A}}^{dat_{\mathcal{K}_l}}|^{-1}$, where $|\cdot|$ represents the size of a set.

3.4.2 Anonymity against Weak Adversary

Since any CCA-secure encryption scheme is used and reverse path identifiers and session identifiers are chosen uniformly and independently at random, the following theorem holds.

Theorem 3.4.1 *$\mathcal{A}^w$ can correctly correlate incoming Data packets from non-compromised producers with the outgoing counterparts at a non-compromised anonymizing router only with negligible probability.*

Proof 3.4.1 (Proof of Theorem 3.4.1) *Suppose that two non-compromised producers $p, p' \in \mathbb{P} \setminus \mathbb{P}_{\mathcal{A}^w}$ independently exchange sequences of secret keys $\mathcal{K}_n = (k_1, \cdots, k_i, \cdots, k_n)$ and $\mathcal{K}'_n = (k'_1, \cdots, k'_i, \cdots, k'_n)$ with their chosen anonymizing routers to build circuits, and $a_i \in \mathbb{A} \setminus \mathbb{A}_{\mathcal{A}^w}$ is used as the i-th anonymizing router in both circuits. Two encrypted Data packets from p and p′ received by a_i are denoted by $dat_{\mathcal{K}_{n-i+1}}$ and $dat'_{\mathcal{K}'_{n-i+1}}$, respectively ($dat, dat' \in \mathbb{D}$). Suppose that $\mathcal{A}^w$ attempts to correlate these incoming Data packets with the corresponding outgoing Data packets, i.e., $dat_{\mathcal{K}_{n-i}}$ and $dat'_{\mathcal{K}'_{n-i}}$. Suppose that an encryption algorithm* Enc *of any CCA-secure secret key encryption scheme Π is used in the encapsulation algorithm $\mathcal{E}$ defined in Section 3.4.1. First, by the definition of CCA-secure encryption schemes, $\mathcal{A}^w$ can correctly correlate them by observing the changes in their bit patterns due to decryption at a_i only with negligible probability. This is because $\mathcal{A}^w$ who does not compromise a_i cannot learn secret keys k_i, k'_i. Second, in ACPNDN, reverse path identifiers and session identifiers, which are the unencrypted parts of content names on Data packets, can be considered as the other source of information to correlate Data packets. The author assumes that $dat_{\mathcal{K}_{n-i+1}}$ and $dat_{\mathcal{K}_{n-i}}$ carry reverse path identifiers rID_i, rID_{i-1} and session identifiers sID_i, sID_{i-1}, respectively. Similarly, $dat'_{\mathcal{K}'_{n-i+1}}$ and $dat_{\mathcal{K}_{n-i}}$ are assumed to carry reverse* *path identifiers rID'_i, rID'_{i-1} and session identifiers sID'_i, sID'_{i-1}, respectively. $\mathcal{A}^w$ can correlate these Data packets if $\mathcal{A}^w$ can correlate any pair of these identifiers on the input and the output Data packet. However, it is infeasible for $\mathcal{A}^w$ because they are chosen uniformly and independently at random by p and p' from $\{0,1\}^\kappa$ as described in Section 3.3.3.*

Next, the author describes the requirement to achieve producer anonymity in ACPNDN.

Theorem 3.4.2 *$p \in \mathbb{P} \setminus \mathbb{P}_{\mathcal{A}^w}$ has producer anonymity in a configuration CF with respect to $\mathcal{A}^w$ if $\exists p' \in \mathbb{P} \setminus \mathbb{P}_{\mathcal{A}^w}$ such that $\mathsf{CF}_{\mathbb{D}}(p') \neq \mathsf{CF}_{\mathbb{D}}(p)$, $p' \neq p$, and any of the following conditions holds:*

- $\mathsf{CF}_{\mathbb{A}}(p) = \mathsf{CF}_{\mathbb{A}}(p') = \perp$.
- $p, p' \in \mathsf{AS}_{\mathcal{A}^w}^{\mathsf{CF}_{\mathbb{D}}(p)_{\mathcal{K}_n}}$.
- $\exists i \in \{1, \cdots, n\}$, $\mathsf{CF}_{\mathbb{A}_i}(p) = \mathsf{CF}_{\mathbb{A}_i}(p') \in \mathbb{A} \setminus \mathbb{A}_{\mathcal{A}^w}$.
- $\exists i \in \{1, \cdots, n\}$, $\mathsf{CF}_{\mathbb{A}_i}(p), \mathsf{CF}_{\mathbb{A}_i}(p') \in \mathbb{A} \setminus \mathbb{A}_{\mathcal{A}^w}$ *and* $\mathsf{CF}_{\mathbb{A}_i}(p), \mathsf{CF}_{\mathbb{A}_i}(p') \in \mathsf{AS}_{\mathcal{A}^w}^{\mathsf{CF}_{\mathbb{D}}(p)_{\mathcal{K}_{n-i}}}$.

In other words, p can anonymously publish their Data packet if there exists another producer p' who publishes another Data packet and any of the following conditions holds: 1. the Data packets of p and p' are returned from caches; 2. p and p' are included in the same sender anonymity set of the encrypted Data packets from p and p' with respect to $\mathcal{A}^w$; 3. p and p' share the same i-th anonymizing router which is not compromised by $\mathcal{A}^w$; or 4. The i-th anonymizing routers of p and p' are not compromised by $\mathcal{A}^w$ and these anonymizing routers are included in the same sender anonymity set of their output Data packets with respect to $\mathcal{A}^w$.

Proof 3.4.2 (Proof of the first condition of Theorem 3.4.2) *Without loss of generality, the author assumes CF such that $\mathsf{CF}(p) = (\perp, c, dat)$ and $\mathsf{CF}(p') = (\perp, c', dat')$, and $\mathcal{A}^w$ such that $\mathbb{P}_{\mathcal{A}^w} = \mathbb{P} \setminus \{p, p'\}$, $\mathbb{A}_{\mathcal{A}^w} = \mathbb{A}$, $\mathbb{R}_{\mathcal{A}^w} = \mathbb{R}$, and $\mathbb{C}_{\mathcal{A}^w} = \mathbb{C}$, i.e., $\mathcal{A}^w$ compromises all the producers except for p and p' and all the anonymizing routers, network routers, and consumers. This CF satisfies the first condition in theorem 3.4.2. In this case, dat and dat′ are published without using circuits. Supposing another configuration CF' which is identical to CF except that $\mathsf{CF}(p) = (\perp, c', dat')$ and $\mathsf{CF}(p') = (\perp, c, dat)$, the only sources of information to distinguish CF and CF' are onion names and signatures on dat and dat′. However, $\mathsf{CF}' \equiv_{\mathcal{A}^w} \mathsf{CF}$ holds because the onion names and signatures are generated from identity key pairs chosen independently and randomly by p and p'. Therefore, from the definition 3.2.3, p has producer anonymity.*

Proof 3.4.3 (Proof of the second condition of Theorem 3.4.2) *Without loss of generality, the author assumes* CF *such that* $\mathsf{CF}(p) = (a_1, \cdots, a_n, c, dat)$ *and* $\mathsf{CF}(p') = (a'_1, \cdots, a'_n, c', dat')$, *and* $\mathcal{A}^w$ *such that* $\mathbb{P}_{\mathcal{A}^w} = \mathbb{P} \setminus \{p, p'\}$, $\mathbb{A}_{\mathcal{A}^w} = \mathbb{A}$, *and* $\mathbb{C}_{\mathcal{A}^w} = \mathbb{C}$, *i.e.,* $\mathcal{A}^w$ *compromises all the producers except for* p *and* p' *and all the anonymizing routers and consumers. The author also assumes that* $p, p' \in \mathsf{AS}_{\mathcal{A}^w}^{dat_{\mathcal{K}_n}}$, *i.e.,* p *and* p' *are included in the same sender anonymity set with respect to* $\mathcal{A}^w$. *This* CF *satisfies the second condition in theorem 3.4.2. The author supposes another configuration* CF' *which is identical to* CF *except that* $\mathsf{CF}'(p) = (a'_1, \cdots, a'_n, c', dat')$ *and* $\mathsf{CF}'(p') = (a_1, \cdots, a_n, c, dat)$. *In this case,* $\mathcal{A}^w$ *can track* dat *and* dat' *from* a_1 *to* a_n *and from* a'_1 *to* a'_n *for* CF *and* CF' *because all the anonymizing routers are compromised. Thus,* $\mathcal{A}^w$ *can distinguish* CF *from* CF' *only if* $\mathcal{A}^w$ *can correctly correlate the input Data packets at* a_1 *and* a'_1 *with* p *and* p'. *However, such* $\mathcal{A}^w$ *cannot exist from the definition of the sender anonymity set. Therefore,* $\mathsf{CF}' \equiv_{\mathcal{A}^w} \mathsf{CF}$ *holds for such* CF', *and* p *has producer anonymity.*

Proof 3.4.4 (Proof of the third condition of Theorem 3.4.2) *Without loss of generality, the author assumes* CF *such that* $\mathsf{CF}(p) = (a_1, \cdots, a_{i-1}, a_i, a_{i+1}, \cdots, a_n, c, dat)$ *and* $\mathsf{CF}(p') = (a'_1, \cdots, a'_{i-1}, a_i, a'_{i+1}, \cdots, a'_n, c', dat')$ *and* $\mathcal{A}^w$ *such that* $\mathbb{P}_{\mathcal{A}^w} = \mathbb{P} \setminus \{p, p'\}$ *and* $\mathbb{A}_{\mathcal{A}^w} = \mathbb{A} \setminus \{a_i\}$, $\mathbb{R}_{\mathcal{A}^w} = \mathbb{R}$, *and* $\mathbb{C}_{\mathcal{A}^w} = \mathbb{C}$, *i.e.,* $\mathcal{A}^w$ *compromises all the producers except for* p, p', *all the anonymizing routers except for* a_i, *and all the network routers and consumers. This* CF *satisfies the third condition in theorem 3.4.2. The author assumes another configuration* CF' *which is identical to* CF *except that* $\mathsf{CF}'(p) = (a_1, \cdots, a_{i-1}, a_i, a'_{i+1} \cdots, a'_n, c', dat')$ *and* $\mathsf{CF}'(p') = (a'_1, \cdots, a'_{i-1}, a_i, a_{i+1}, \cdots, a_n, c, dat)$. *In this case,* $\mathcal{A}^2$ *can track* dat *and* dat' *from* a_1 *to* a_{i-1}, *from* a_{i+1} *to* a_n, *from* a'_1 *to* a'_{i-1}, *and from* a'_{i+1} *to* a'_n *for* CF *and* CF'. *Thus,* $\mathcal{A}^w$ *can distinguish* CF *from* CF' *only if* $\mathcal{A}^w$ *can correctly correlate the inputs and outputs at* a_i. *However, such* $\mathcal{A}^w$ *does not exist because it contradicts theorem 3.4.1. Therefore,* $\mathsf{CF}' \equiv_{\mathcal{A}^w} \mathsf{CF}$ *holds for such* CF', *and* p *has producer anonymity.*

Proof 3.4.5 (Proof of the fourth condition of Theorem 3.4.2) *Without loss of generality, the author assumes* CF *such that* $\mathsf{CF}(p) = (a_1, \cdots, a_{i-1}, a_i, a_{i+1}, \cdots, a_n, c, dat)$ *and* $\mathsf{CF}(p') = (a'_1, \cdots, a'_{i-1}, a'_i, a'_{i+1}, \cdots, a'_n, c', dat')$ *and* $\mathcal{A}^w$ *such that* $\mathbb{P}_{\mathcal{A}^w} = \mathbb{P} \setminus \{p, p'\}$ *and* $\mathbb{A}_{\mathcal{A}^w} = \mathbb{A} \setminus \{a_i, a'_i\}$, *and* $\mathbb{C}_{\mathcal{A}^w} = \mathbb{C}$, *i.e.,* $\mathcal{A}^w$ *compromises all the producers except for* p, p', *all the anonymizing routers except for* a_i, a'_i, *and all the consumers. The author also assumes that* $a_i, a'_i \in \mathsf{AS}_{\mathcal{A}^w}^{dat_{\mathcal{K}_{n-i}}}$, *i.e.,* a_i *and* a'_i *are included in the same sender anonymity set of their output Data packets with respect to* $\mathcal{A}^w$. *This* CF *satisfies the fourth condition in theorem 3.4.2. The author assumes another configuration* CF' *which is identical to* CF *except that* $\mathsf{CF}'(p) = (a_1, \cdots, a_{i-1}, a_i, a'_{i+1}, \cdots, a'_n, c', dat')$

and $\mathsf{CF}'(p') = (a'_1, \cdots, a'_{i-1}, a'_i, a_{i+1}, \cdots, a_n, c, dat)$*. In this case,* $\mathcal{A}^w$ *can track dat and dat′ from* a_1 *to* a_{i-1}*, from* a_{i+1} *to* a_n*, from* a'_1 *to* a'_{i-1}*, and from* a'_{i+1} *to* a'_n *for* CF *and* CF'*. Thus,* $\mathcal{A}^w$ *can distinguish* CF *and* CF' *only if* $\mathcal{A}^w$ *can correctly correlate* a_i*,* a'_i *with their output Data packets. However, such* $\mathcal{A}^w$ *cannot exist from the definition of the sender anonymity set. Therefore,* $\mathsf{CF}' \equiv_{\mathcal{A}^w} \mathsf{CF}$ *holds for such* CF'*, and p has producer anonymity.*

In hidden service, receiver anonymity is provided against $\mathcal{A}^w$ if a circuit includes at least one non-compromised anonymizing router. In contrast, as shown in Theorem 3.4.2, ACPNDN provides anonymity even if all the anonymizing routers in a circuit are compromised because a producer can be included in a sender anonymity set with respect to $\mathcal{A}^w$ who does not compromise the producer's first-hop network router. This corresponds to the second condition. The fourth condition is similar to the second condition. In contrast to the fact that each anonymizing router can learn its predecessor from the source address in a received packet in hidden service, each anonymizing router cannot learn its predecessor in ACPNDN. Thus, the fourth condition implies that it is more difficult for $\mathcal{A}^w$ to compromise a circuit in ACPNDN than hidden service. The first condition implies that producer anonymity can be provided by leveraging in-network caching. This is because producers do not send/receive any packets when cache hits occur.

The inverse of Theorem 3.4.2 also holds: producer p does not have producer anonymity if none of the conditions of theorem 3.4.2 hold, i.e., Data packet *dat* of p is not returned from caches and $\mathcal{A}^w$ compromises the first-hop network router of p and all the anonymizing routers in p's circuit. This is because it implies that $\mathcal{A}$ can track *dat* throughout the circuit, i.e., the circuit is compromised by $\mathcal{A}^w$. Let $f_{\mathbb{A}} = |\mathbb{A}_{\mathcal{A}}|/|\mathbb{A}|$ and $f_{\mathbb{R}} = |\mathbb{R}_{\mathcal{A}}|/|\mathbb{R}|$ $(0 \leq f_{\mathbb{A}}, f_{\mathbb{R}} < 1, \mathcal{A} \in \{\mathcal{A}^w, \mathcal{A}^s\})$, i.e., $f_{\mathbb{A}}$ and $f_{\mathbb{R}}$ represent the fractions of compromised entities in the sets of all the anonymizing routers and the network routers, respectively. The author assumes that each anonymizing router in a circuit is compromised independently with probability $f_{\mathbb{A}}$. This gives a good approximation in the realistic model, where a large number of anonymizing routers exist (e.g., $|\mathbb{A}| \approx 6000$ in current Tor [84]). Then, $\mathcal{A}^w$ can break producer anonymity with probability

$$(1 - p_c) \cdot f_{\mathbb{R}} \cdot f_{\mathbb{A}}^n, \tag{3.1}$$

where p_c is the probability that a cache hit occurs $(0 \leq p_c \leq 1)$. Intuitively, larger n, i.e., longer circuits, contributes to achieve producer anonymity with more confidence. In contrast, an adversary againt hidden service can break receiver anonymity with probability $f_{\mathbb{A}}^n$ $(\geq (1 - p_c) \cdot f_{\mathbb{R}} \cdot f_{\mathbb{A}}^n)$.

From the above analysis, the author concludes that ACPNDN provides more resilient anonymity than hidden service against $\mathcal{A}^w$ when the same number of anonymizing routers are used. In particular, ACPNDN provides a level of anonymity with one fewer anonymizing router than hidden service if $f_{\mathbb{A}} \geq f_{\mathbb{R}}$ holds. The author argues that this condition holds in practical scenarios because anonymizing routers are held by volunteers and thus an adversary can easily prepare a set of compromised anonymizing routers.

3.4.3 Anonymity against Strong Adversary

$\mathcal{A}^s$ can launch more sophisticated attacks called *traffic analysis attacks* [25,50–53] by using other sources of information, such as timing and volume of packets. In particular, the author focuses on *traffic confirmation attacks*, which are a type of traffic analysis attacks aiming to correlate two entities included in the same circuit. To mitigate traffic confirmation attacks, several schemes like packet batching and reordering have been proposed [47,53], however, they are not implemented in Tor and hidden service due to their high cost in terms of the delay and the load at each anonymizing router. Thus, ACPNDN does not explicitly employ such schemes, whereas the author believes that they could also be incorporated into ACPNDN almost without modification.

Rather than discussing how to deal with traffic confirmation attacks, the author analyzes the probability that producer anonymity is broken with the predecessor attack [25,85], assuming that $\mathcal{A}^s$ launches successful traffic confirmation attacks. The predecessor attack is a substantial threat to anonymity systems based on onion routing because it is difficult to detect, relatively easy to launch, and the probability of its success increases to 1.0 with time [25]. With the predecessor attack, $\mathcal{A}^s$ can break producer anonymity even when $\mathcal{A}^w$ cannot.

Predecessor attack in static model

Before describing the predecessor attack, the author describes the cases where producer anonymity and receiver anonymity are broken with traffic confirmation attacks and show the probabilities of their occurrence in a single round. The author assumes that traffic confirmation attacks always succeeds when two entities on the same circuit are compromised by $\mathcal{A}^s$. In the current implementation of hidden service, each circuit of receivers includes three anonymizing routers by default. To match the level of anonymity against $\mathcal{A}^w$, the author assumes that two anonymizing routers are included in each circuit in ACPNDN.

The upper illustration in Figure 3.6 depicts the case where anonymity of a producer is broken in ACPNDN. AR and RP are an anonymizing router and a rendezvous point included in the producer's

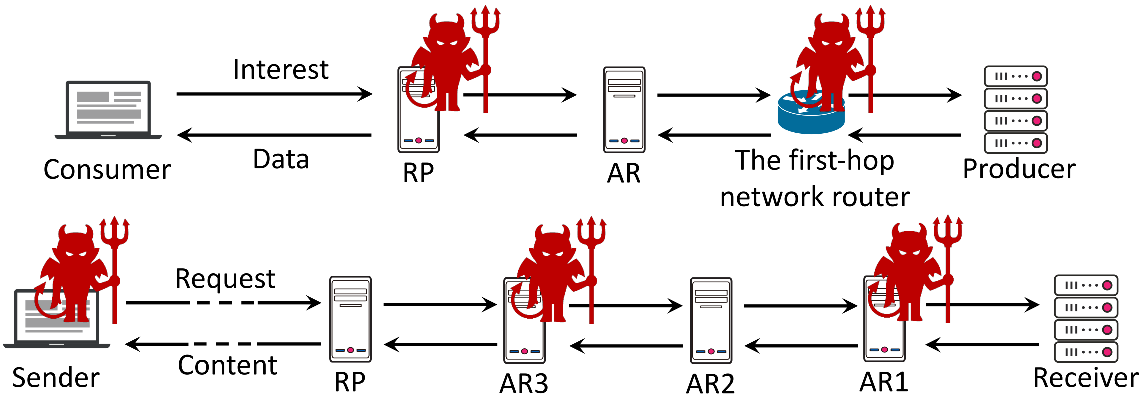


Figure 3.6: The cases where receiver anonymity and producer anonymity are broken in ACPNDN (upper illustration) and in hidden service (lower illustration) with traffic confirmation attacks.

circuit, respectively. The author focuses only on the case where Interest packets are not satisfied by caches because $\mathcal{A}^s$ cannot launch traffic confirmation attacks if cache hits occur. In ACPNDN, producer anonymity is broken with traffic confirmation attacks if both the first-hop network router and the rendezvous point of the producer's circuit are compromised by $\mathcal{A}^s$. This is because it implies that $\mathcal{A}^s$ can correlate the producer's MAC address obtained at the first-hop network router and plaintext Data packets outputted by the rendezvous point. Note that $\mathcal{A}^s$ can make sure that it has compromised a rendezvous point among anonymizing routers because packets are plaintext only between a consumer and a rendezvous point. Similarly, $\mathcal{A}^s$ can make sure that a compromised network router is the first-hop of a producer by checking whether the packets received on the network router are the same as those received on the compromised rendezvous point. If not the same, there is an anonymizing router between the network router and the rendezvous point and thus the network router is the first-hop of a producer. Thus, producer anonymity is broken in ACPNDN with probability $f_{\mathbb{R}} \cdot f_{\mathbb{A}}$.

Similarly, the lower illustration in Figure 3.6 depicts one of the cases where anonymity of a receiver is broken in hidden service. Note that the last-hop anonymizing router of a receiver's circuit is not the rendezvous point in hidden service. Instead, the last-hop anonymizing router of a sender's circuit is the rendezvous point. Traffic confirmation attacks against hidden service looks a little more complicated due to end-to-end encryption between senders and receivers: $\mathcal{A}^s$ requires acting as a sender to observe packets from a receiver in plaintext. However, this requirement is easily satisfied because anyone can act as a sender and thus the probability that a sender is compromised can be ignored. The goal of $\mathcal{A}^s$ acting as a sender is to learn a receiver's IP address at the first-hop anonymizing router. Even if $\mathcal{A}^s$ finds that a compromised anonymizing router is included in a receiver's circuit, however, $\mathcal{A}^s$ cannot confirm its position in the circuit. Thus, in addition to the first-hop anonymizing router (AR1), $\mathcal{A}^s$ must compromise another anonymizing router in the same circuit (AR2 or AR3). Then, $\mathcal{A}^s$ makes sure of their positions in a circuit based on the IP addresses of the compromised anonymizing routers and the rendezvous point known to the sender (i.e., $\mathcal{A}^s$). For example, if $\mathcal{A}^s$ learns that two compromised anonymizing routers in the same circuit are not adjacent to each other and one of them is adjacent to the rendezvous point, then these anonymizing routers are AR1 and AR3. Therefore, receiver anonymity is broken in hidden service with probability $f_{\mathbb{A}} \cdot (1 - (1 - f_{\mathbb{A}})^2) = 2f_{\mathbb{A}}^2 - f_{\mathbb{A}}^3$.

In the predecessor attack, $\mathcal{A}^s$ just acts like legitimate entities, i.e., follows the prescribed protocols for the compromised entities, and continually performs a succession of traffic confirmation attacks against the packets passing through them [25]. In this subsection, the author considers the predecessor attack in the static model, where the entities in $\mathbb{A}$ and $\mathbb{R}$ do not change throughout rounds. In each round, a circuit is compromised by $\mathcal{A}^s$ in hidden service and in ACPNDN with probabilities of $2f_{\mathbb{A}}^2 - f_{\mathbb{A}}^3$ and $f_{\mathbb{R}} \cdot f_{\mathbb{A}}$, respectively as described above. These are the lower bound of the probabilities of a circuit being compromised with the predecessor attack. Comparing to Eq. (3.1), the author can see that the predecessor attack has a greater probability of success in many cases. Because receiver and producer anonymity are broken if one circuit is compromised, the author focuses on the probability that at least one circuit is compromised by $\mathcal{A}^s$ in m rounds, hereinafter.

If all the anonymizing routers in circuits are chosen uniformly at random in each round, the probability that at least one circuit of a receiver is compromised in m rounds in hidden service is derived as follows:

$$1 - (1 - 2f_{\mathbb{A}}^2 + f_{\mathbb{A}}^3)^m. \tag{3.2}$$

As the number of rounds increases ($m \to \infty$), the probability grows to 1.0.

To mitigate the predecessor attack, entry guards are introduced to hidden service. Because a receiver repeatedly uses the first-hop anonymizing router called an entry guard, $\mathcal{A}^s$ must compromise it in addition to either the second-hop or the third-hop anonymizing router. The probability that neither the second-hop nor the third-hop anonymizing router is compromised even once in m rounds is $(1 - f_{\mathbb{A}})^{2m}$. Therefore, the probability that at least one circuit of a receiver is compromised is derived as follows:

$$f_{\mathbb{A}} \cdot \{1 - (1 - f_{\mathbb{A}})^{2m}\}. \tag{3.3}$$

As the number of rounds increases ($m \to \infty$), the probability grows to $f_{\mathbb{A}}$ (< 1).

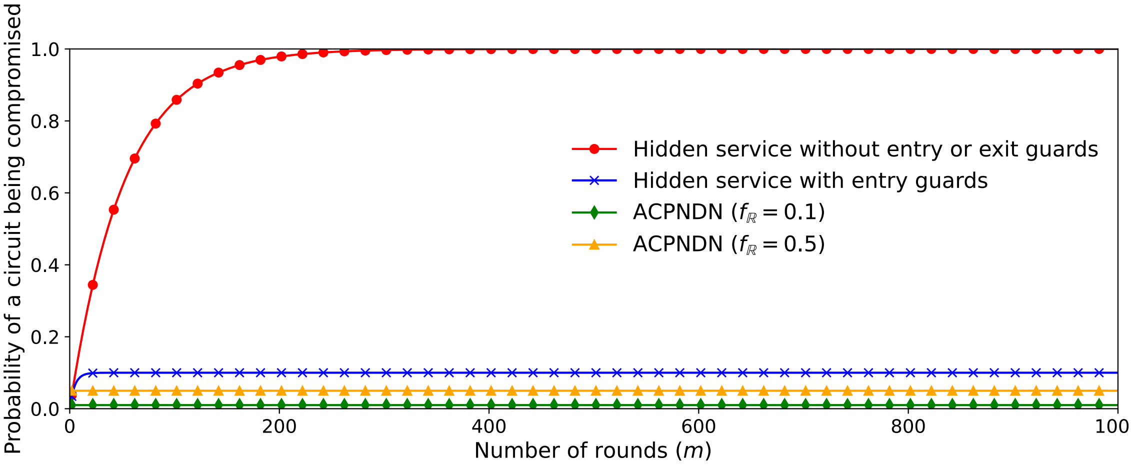


Figure 3.7: The probability that at least one circuit is compromised with the predecessor attack in the static model ($f_{\mathbb{A}} = 0.1$).

In ACPNDN, the last-hop anonymizing router in circuits of a producer, i.e., a rendezvous point, is fixed. Such an anonymizing router is called an exit guard. In addition, the first-hop network router plays the role of an entry guard because the producer's identities, such as MAC addresses, are revealed only to it. Thus, $\mathcal{A}^s$ must compromise both of them to compromise a circuit. Therefore, the probability that at least one circuit of a producer is compromised is derived as follows:

$$f_{\mathbb{R}} \cdot f_{\mathbb{A}}, \tag{3.4}$$

regardless of the number of rounds ($f_{\mathbb{R}} \cdot f_{\mathbb{A}} < f_{\mathbb{A}} < 1.0$). This probability is equal to the lower bound of probability of a circuit being compromised by the predecessor attack in the static model.

Figure 3.7 shows the changes in Eq. (3.2), Eq. (3.3), and Eq. (3.4) as the number of rounds increases when $f_{\mathbb{A}} = 0.1$. It is shown that ACPNDN offers the best security even if $f_{\mathbb{R}}$ is much larger than $f_{\mathbb{A}}$.

Predecessor attack in dynamic model

In this subsection, the author considers the predecessor attack in the dynamic model, where the members in $\mathbb{A}$ and $\mathbb{R}$ change over rounds. As distinct from the static model, changes of an entry and an exit guard are caused if one of them becomes unavailable, and this gives $\mathcal{A}^s$ further opportunities to compromise circuits in addition to those in the static model. The dynamic model is worth considering because it has been shown that only about half of the entry guards in hidden service remain available for the intended period of time (e.g., 720-1440 hours) and thus changes of guards occur when they are undesirable [86]. Since anonymizing routers are operated by unreliable volunteers, the author believes that this is a inherent problem for hidden service and ACPNDN.

It is possible to use both entry and exit guards in hidden service to improve security against the predecessor attack in the static model. However, this causes a problem in the dynamic model because the receiver changes entry and exit guards if one of them becomes unavailable and thus changes of guards are presumed to occur more frequently in the case where both entry and exit guards are used than in the case where only entry guards are used [85]. Although ACPNDN employs both entry and exit guards, ACPNDN mitigates this problem by having the first-hop network routers act as entry guards instead of the first-hop anonymizing routers. Because (carrier-grade) network routers managed by ISPs are intended to provide higher availability, e.g., the five nines available requirement [87], than anonymizing routers operated by voluntary hosts, changes of guards are affected almost exclusively by the availability of anonymizing routers chosen as exit guards. This implies that ACPNDN provides a degree of security against the predecessor attack in the dynamic model that is equivalent to that provided by hidden service, where only entry guards are employed.

In the following description, the author compares the case where anonymizing routers are used as an entry and an exit guard in hidden service with ACPNDN, where a network router and an anonymizing router are used, in terms of the probability that at least one circuit is compromised by $\mathcal{A}^s$ in m rounds. The goal of this subsection is to show how the probability decreases by substituting network routers for anonymizing routers. The author assumes that each anonymizing router becomes unavailable in $\mathbb{A}$ independently with probability q $(0 < q < 1)$ at the end of each round. Because the author is interested in how the change in q affects the probability of a circuit being compromised, the author assumes $f = f_{\mathbb{A}} = f_{\mathbb{R}}$ $(0 \leq f < 1)$, i.e., the same fraction of anonymizing routers and network routers are compromised by $\mathcal{A}$.

When anonymizing routers are used as both entry and exit guards in hidden service, the probability that the producer changes guards i times in m rounds is $\{1-(1-q)^2\}^i\{(1-q)^2\}^{m-i}\binom{m}{i}$. For i changes of guards, the probability that $\mathcal{A}^s$ succeeds in compromising at least one circuit of the producer is $1-(1-2f^2+f^3)^i$. Thus, the probability that $\mathcal{A}^s$ succeeds in compromising at least one circuit of the producer in m rounds is derived as follows:

$$\sum_{i=0}^{m}\{1-(1-2f^2+f^3)^i\}\{1-(1-q)^2\}^i\{(1-q)^2\}^{m-i}\binom{m}{i}. \tag{3.5}$$

Next, the author considers ACPNDN, where network routers and anonymizing routers are used

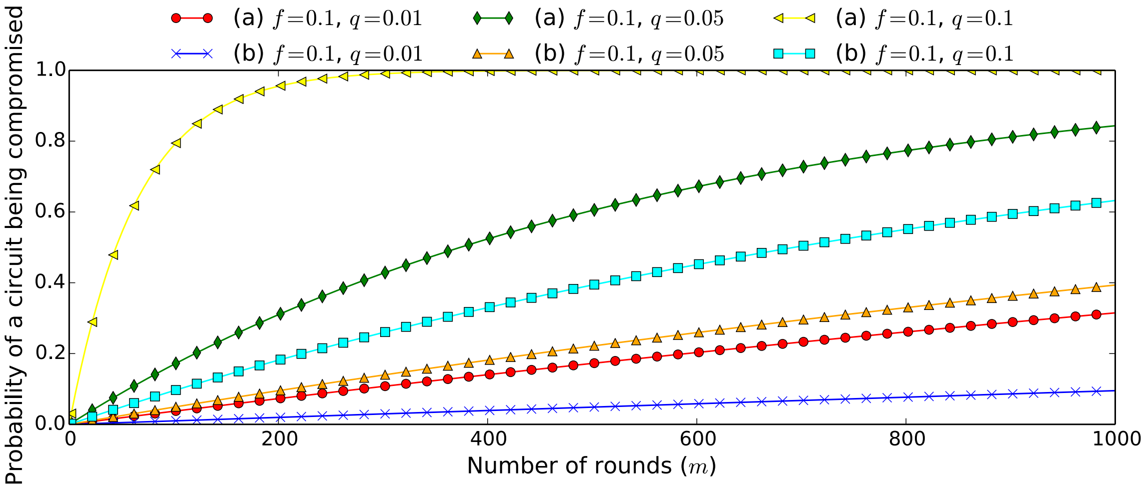


Figure 3.8: Probabilities that at least one circuit is compromised with the predecessor attack in the dynamic model. (a) represents the case where both entry and exit guards are used in hidden service (Eq. (3.5)), and (b) represents ACPNDN (Eq. (3.6)).

as entry and exit guards, respectively. The author assumes that the probability that each network router becomes unavailable in $\mathbb{R}$ at the end of each round is sufficiently close to 0.0. Therefore, in ACPNDN, the probability that the producer changes guards i times in m rounds is $q^i(1-q)^{m-i}\binom{m}{i}$. For i changes of guards, the probability that $\mathcal{A}^s$ succeeds in compromising at least one circuit of the producer is $1-(1-f^2)^i$ since $\mathcal{A}^s$ must compromise the first-hop network router and the last-hop anonymizing router. Thus, the probability that $\mathcal{A}^s$ succeeds in compromising at least one circuit of the producer in m rounds is derived as follows:

$$\sum_{i=0}^{m}\{1-(1-f^2)^i\}q^i(1-q)^{m-i}\binom{m}{i}. \tag{3.6}$$

Figure 3.8 shows the change in Eq. (3.5) and Eq. (3.6) for several pairs of f and q. It is shown that the probabilities of a circuit being compromised are sufficiently small in ACPNDN for all the pairs of f and q and the differences increase as q increases.

From the analysis of both models, the security of ACPNDN against the predecessor attack can be summarized as follows: First, ACPNDN provides the better security in the static model because both entry and exit guards are used. Second, ACPNDN provides the security comparable to that of hidden service in the dynamic model due to the use of network routers as entry guards. Because $\mathcal{A}^s$ can launch the predecessor attack in the static model and the dynamic model at the same time, the author concludes that ACPNDN provides better security against the predecessor attack than hidden service.

3.5 Evaluation

This section first evaluates the performance of ACPNDN compared to hidden service in terms of RTT, defined as the time it takes for a consumer to send a request and then receive the corresponding the content, and throughput by implementing their prototypes. ACPNDN is only compared with hidden service since no other anonymity systems are proposed for NDN which assume the same adversarial model as ACPNDN. The author then assesses the probability of the successful predecessor attack assuming $\mathcal{A}^s$ with the knowledge of the underlying network topology to address a problem on realistic networks.

3.5.1 Implementation and Performance

This subsection focuses on the content publication phase because the same style of communication is used in the other phases and they have little effect on producers' long-term activities, owing to the fact that they are performed only at the first set up time. In addition, the author assumes that Interest packets issued by a consumer are forwarded to a producer without being satisfied by network routers. This is the worst-case scenario in terms of RTT and throughput.

The author implemented ACPNDN as applications run on producers and anonymizing routers (including rendezvous points) by using the ndn-cxx library, which is a C++ library implementing NDN primitives. These applications implement the functions required in the content publication phase, such as encryption and decryption of packets, described in Section 3.3.5. The author used AES-128 as a secret key encryption/decryption algorithm and HMAC with SHA-256 as a message authentication code generation/verification algorithm. These cryptographic functions were implemented by using OpenSSL. To compare ACPNDN with hidden service, the author also implemented simple hidden service applications that work as receivers and anonymizing routers because the current hidden service implementation includes many functions not required for comparison. In hidden service, the author assumes that a sender issues an IP packet requesting content through a circuit built by a receiver without using a circuit between a sender and a rendezvous point so that only receiver anonymity comparable to producer anonymity is provided.

To derive RTT, the author focuses mainly on the process delay taken by the applications. This is because current NDN runs as an application on top of TCP/IP, and thus an unavoidable overhead will be incurred in ACPNDN, which is implemented over NDN, compared to hidden service, which is implemented over TCP/IP, when producers/receivers and anonymizing routers communicate with each other, as described in [16, 69]. The author first evaluates the overall process delay taken by the

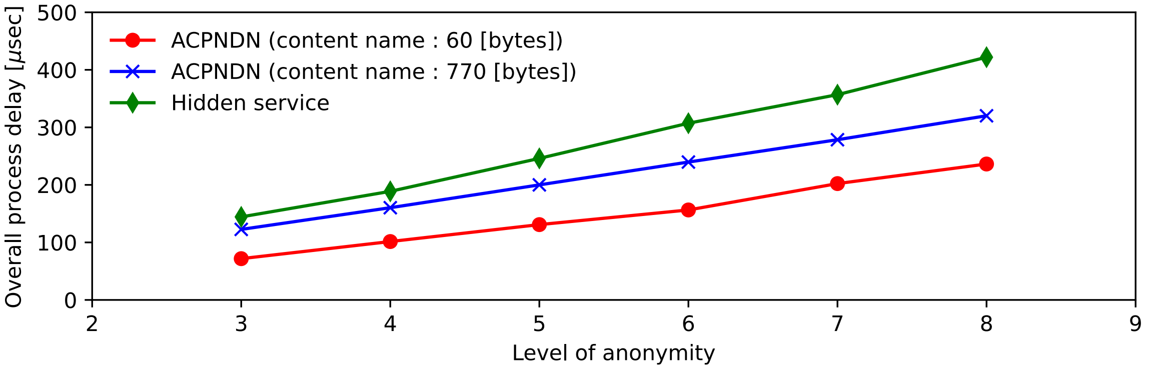


Figure 3.9: Process delay measurement.

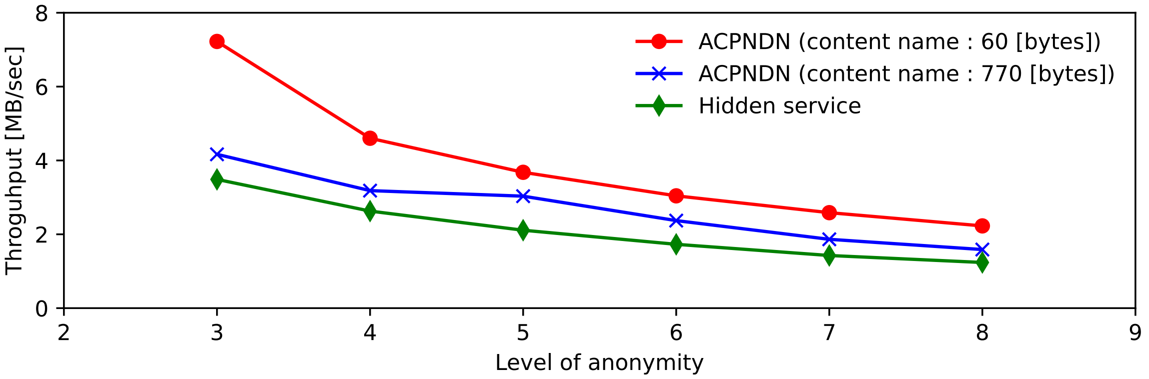


Figure 3.10: Throughput measurement.

applications of ACPNDN and hidden service and then derives the RTT for various propagation delays between neighboring entities of a circuit, assuming a simple line topology. The overall process delay is defined as the total time it takes for the applications to process a packet by using cryptographic functions, not including the time it takes to transport packets between them. All experiments were conducted on a machine with an Intel(R) Xeon(R) E5-2620 v4 processor (2.10 GHz) with eight DDR4 16GB DRAM devices. The operating system is Ubuntu 18.04 LTS.

Figure 3.9 shows the overall process delay as a function of the level of anonymity provided against $\mathcal{A}^w$. Similarly, Figure 3.10 shows the throughput. A level of anonymity is defined as the number of anonymizing routers and network routers $\mathcal{A}^w$ must compromise to trace packets throughput a circuit; for example, if the level of anonymity is three, it means that three anonymizing routers are used in hidden service and that two anonymizing router is used in ACPNDN. Because the size of application data is set to 512 bytes in the current implementation of hidden service, the author sets the payload size of a Data packet to 512 bytes. Regarding the size of a content name in Interest/Data packets,

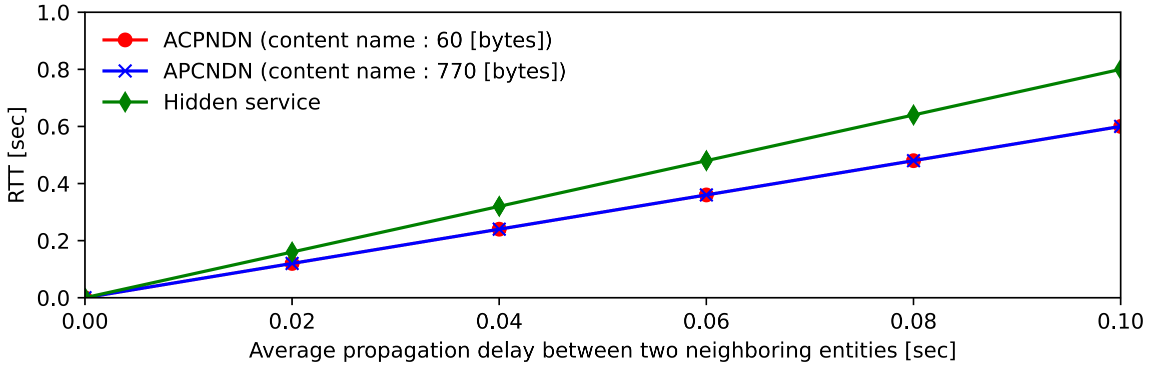


Figure 3.11: RTT measurement (level of anonymity = 3).

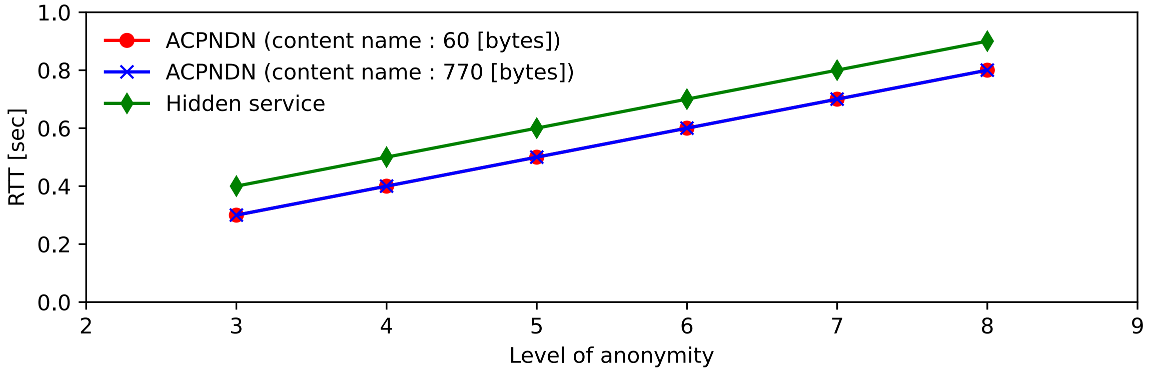


Figure 3.12: RTT measurement (average propagation delay = 0.05 [sec]).

Ghali et al. generated realistic NDN compatible names according to the URLs in the Unibas dataset from The Content Name Collection [74] and found that the average and the maximum size of the names are approximately 60 bytes and 770 bytes, respectively. According to this result, the author set the size of content names to these sizes.

As expected, ACPNDN has better performance (i.e., less process delay and more throughput) for shorter content names. This is mainly because the shorter content names result in shorter process delay for running the cryptographic operations and generating packets. This implies that producers can improve efficiency of content publishing by carefully naming their content in ACPNDN. In comparison with hidden service, ACPNDN has better performance because ACPNDN reduces the number of required anonymizing routers in a circuit to achieve a comparable level of anonymity to hidden service by one. Specifically, ACPNDN reduces the number of the cryptographic operations performed by a producer compared to those performed by a receiver in hidden service, in addition to that one anonymizing router becomes unnecessary.

Figure 3.11 shows how the propagation delay between neighboring entities of a circuit affects the RTT under the condition that the level of anonymity is three (i.e., hidden service uses three anonymizing routers and ACPNDN uses two anonymizing routers). The author assumes that the average propagation delay is a few tens of milliseconds because a circuit should be composed of anonymizing routers distributed all over the world to minimize the probability that all the anonymizing routers of a consumer's choice are compromised by an adversary. The author concludes that RTT is predominantly determined by the propagation delay required to transport packets through geographically distributed anonymizing routers. In contrast, the overhead due to the process delay is a few hundreds of microseconds, as shown in Figure 3.9, and thus causes negligible effect. The overhead due to the propagation delay is unavoidable to provide anonymity; however, the evaluation shows that ACPNDN effectively prunes RTT (about 0.1 [sec]) by reducing the number of anonymizing routers by one. Similarly, Figure 3.12 show how the level of anonymity affects the RTT, assuming that the average propagation delay is 0.05 [sec]. As expected, the RTT increases in proportion to the level of anonymity. Therefore, each producer can trade off the level of their anonymity and RTT in ACPNDN by carefully determining the number of anonymizing routers in their circuit.

3.5.2 Anonymity Analysis under a Realistic Network Topology

The author analyzed the security of ACPNDN against the predecessor attack launched by $\mathcal{A}^s$ in Section 4.3.3, assuming the adversarial model where $\mathcal{A}^s$ chooses network routers to compromise uniformly at random. This subsection analyzes the security against the predecessor attack against $\mathcal{A}^s$ who chooses anonymizing routers to compromise taking the underlying network topology into account. Because the security of ACPNDN depends on the first-hop network routers of producers, $\mathcal{A}^s$ can break anonymity of many producers if it preferentially compromises network routers connecting to many ones. Thus, the expected number of such producers can depend on an underlying network topology. To confirm this, the author evaluates the expected value of the number of producers being broken anonymity with the predecessor attack in the static model based on the real network topology and population in Tokyo, Japan.

The target area is a 32 km square part of Tokyo. The author constructs a router-level topology defined as a tree of depth 3 based on the positions and coverage areas of telephone exchange buildings of NTT East Corporation [88]. The first-level network router is placed in Otemachi, which is one of the most thickly populated areas in Japan. The second level network routers are placed near the 6 terminal stations on the Yamanote line. The third-level network routers are placed so that the target area is uniformly covered and connected to the closest second level network routers. The fourth level

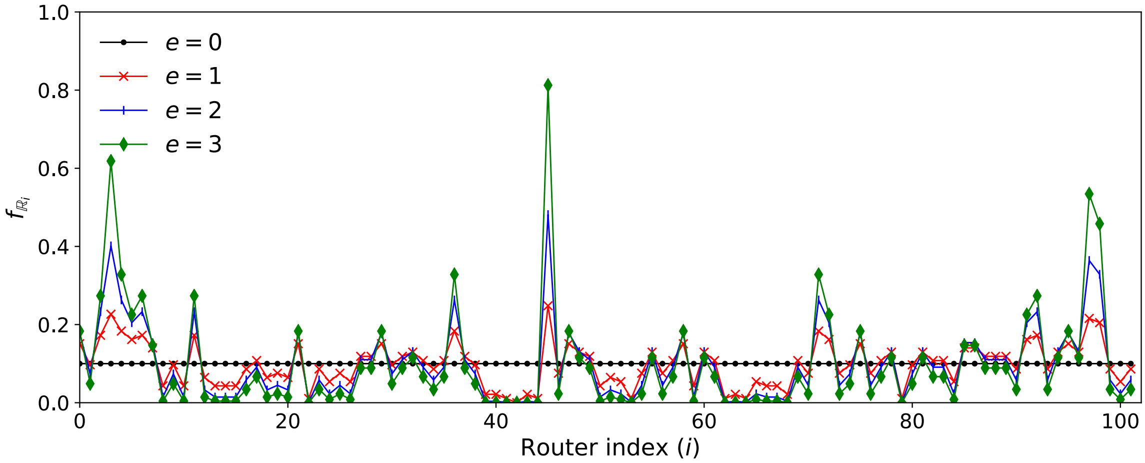


Figure 3.13: Probability that the i-th first-hop network router is compromised.

network routers connect to the hosts within some specific municipalities, and thus plays the role of the first-hop network routers of producers. The fourth-level network routers are referred to as the first-hop network routers, hereinafter. The number of the first-hop network routers is $M = 102$. The author assumes that the number of producers connected to each of the first-hop network routers is proportional to the population within its coverage area, and the number of all producers is 927 (1/10000 of the total population in the target area).

Let n_i and $f_{\mathbb{R}_i}$ denote the number of producers connecting to the i-th first-hop network router and the probability of the i-th first-hop network router being compromised, respectively. The author sets $f_{\mathbb{R}_i}$ to be proportional to $n_i^e / \sum_{j=1}^{M} n_j^e$ so that first-hop network routers with many producers are more likely to be compromised. e is a constant that determines how much priority is given to the network routers with many producers. The author assumes that $\mathcal{A}^s$ compromises ten of the first-hop network routers (i.e., approximately 10 % of all the first-hop network routers), and thus $f_{\mathbb{R}_i} = 0.1$ for all the first-hop network routers when $e = 0$. By setting $f_{\mathbb{R}_i}$ so that $\sum_{i=1}^{M} f_{\mathbb{R}_i}$ is constant for $e \geq 0$, the larger e becomes, the larger $f_{\mathbb{R}_i}$ for the first-hop network routers with larger n_i becomes and the smaller $f_{\mathbb{R}_i}$ for those with smaller n_i becomes. Figure 3.13 shows the values of $f_{\mathbb{R}_i}$ when $e = \{0, 1, 2, 3\}$. For example, the 45-th first-hop network router is compromised with higher probability than others when $e \geq 1$ because it connects to more producers.

As described in Section 3.4.3, the probability that anonymity of each producer connected to the i-th first-hop network router is broken with the predecessor attack in the static model is $f_{\mathbb{R}_i} \cdot f_{\mathbb{A}}$, where $f_{\mathbb{A}}$ is the probability of each anonymizing routers being compromised. Thus, the expected value of the number of producers of which anonymity is broken among all the producers connected to the i-th

Table 3.2: The Expected Value of The Number of Producers Being Broken Anonymity and Its Ratio

	$e = 0$	$e = 1$	$e = 2$	$e = 3$
E	9.01	11.84	13.63	15.06
ratio $(= E/927)$	0.0097	0.0128	0.0147	0.0162

first-hop network router is $\sum_{j=1}^{n_j} j \cdot f_{\mathbb{R}_i} \cdot \binom{m}{i} \cdot f_{\mathbb{A}}^{j} \cdot (1 - f_{\mathbb{A}})^{n_i - j}$. By using linearity of expectation, the expected value of the total number of producers who are broken anonymity is computed as follows:

$$E = \sum_{i=1}^{M} \sum_{j=1}^{n_j} j \cdot f_{\mathbb{R}_i} \cdot \binom{m}{i} \cdot f_{\mathbb{A}}^{j} \cdot (1 - f_{\mathbb{A}})^{n_i - j}. \tag{3.7}$$

The actual values of E and its ratio among all the producers are summarized in Table. 3.2. The result shows that $\mathcal{A}^s$ can break anonymity of more producers by selectively compromising the first-hop network routers connecting to many producers. Thus, it is important for producers who wish to enjoy anonymity to contract ISPs with a small number of hosts if $\mathcal{A}^s$ with the knowledge of the underlying network topology should be contemplated. However, if there are too few hosts in the same network, anonymity is difficult to achieve in principle. Therefore, producers should choose ISPs to contract while simultaneously considering these two factors.

3.6 Discussion and Conclusion

This chapter first defined producer anonymity in terms of content-producer unlinkability by using the notion of configurations to capture the session-less feature of NDN. The author then designed ACPNDN as the first system to provide producer anonymity under a realistic adversary model where adversaries can eavesdrop packets on arbitrary points on the network. The author leveraged onion names and rendezvous points of hidden service to address the problem of NDN that every content is inherently bound to its producer with the producer name and signature. In addition, the author improved hidden service in terms of resiliency against a weak and a strong adversary by carefully incorporating the RICE protocol into the onion routing technique so that network-layer anonymity provided by producer-side edge routers are leveraged. The theoretical and empirical analysis showed that ACPNDN can reduces RTT for content retrieval by reducing the number of anonymizing routers required to achieve a certain level of anonymity by one. Since RTT for content retrieval is a few hundreds of milliseconds in realistic scenarios, the author argues that ACPNDN is suitable for applications where there are no strict time constraints (e.g., web browsing and voice/video chatting).

The author's future research plans include designing a mechanism for balancing anonymity and accountability. On the current Internet, anonymity systems like Tor and hidden services are sometimes used by adversaries to take malicious actions while concealing their identities; for example, an adversary can conceal the command and control (C&C) server sending commands to botnets by using Tor. To address this problem, some existing studies has proposed a method to detect traffic between C&C servers and botnets by analyzing traffic patterns on the Tor network; however, the C&C servers might not be identified even if malicious traffic is detected because they communicate with botnets through multiple anonymizing routers. Therefore, incorporating into ACPNDN the capability to de-anonymize the source of traffic and stop the use of the system if and only if malicious traffic is detected is an important research topic. In terms of security of ACPNDN, integrating several DoS mitigation mechanisms into ACPNDN is also a promising future research plan. For example, requiring producers to solve puzzles, which cost a lot of CPU cycles or memory before establishing reverse paths and circuits, can hinder adversaries from making network routers and anonymizing routers unavailable by establishing many reverse paths and circuits through them.

Chapter 4

Private Retrieval of Location-related Content Using K-anonymity and its Application to NDN

This chapter designs PLCR to provide location privacy for the users of the NDN-based IoT platforms. The rest of the chapter is organized as follows: Section 4.1 describes the system, adversary, and privacy model. Section 4.2 describes the design of PLCR. Section 4.3 proposes a heuristic algorithm for generating location anonymity sets. Section 4.4 evaluates the performance of PLCR and security of the algorithm. Section 4.5 concludes this chapter.

4.1 System, Adversary, and Privacy Model

Table 4.1 summarizes the notation used throughout this chapter.

4.1.1 System Model

Location

IoT is expected to be applied to various scenarios, such as smart homes, smart cars, and building management systems. IoT devices and IoT gateways that hold location-related content are referred to as *producers* and assume that they are installed within a region called a *service region*, denoted by $\mathcal{L}$. Similar to previous studies [40, 63], the service region is divided into arbitrary equal-sized $m \in \mathbb{N}$ squares, each of which is defined as location l_i (i.e., $\mathcal{L} = \{l_1, \cdots, l_m\}$). Each location l_i has a

Table 4.1: Notation.

Notation	Description
$\mathcal{L}$	Service region (set of all locations)
$\mathcal{L}_0$	Set of all locations that cannot be chosen as LOIs
l_i	Location
n_i	Location name
t_i	Content type name
σ_{sk_i}	Signature generated with a secret key sk_i
$I(n_i, t_j)$	Interest packet
$D(n_i, t_j, \sigma_{sk_i})$	Data packet
m	Number of locations in the service region
$batch$	Number of requests and replies an anonymizer accumulates
$delay$	Maximum time a packet is delayed at an anonymizer
$\mathcal{S}, \mathcal{S}_i$	Location anonymity set
$id(\mathcal{S})$	Identifier of a location anonymity set $\mathcal{S}$
$e_{i,\mathcal{S}}$	Variable that indicates whether location l_i can be the LOI of $\mathcal{S}$
p_i	Estimate of the popularity of l_i (i.e., $p_i = \Pr[X_l = l_i]$)
k	Degree of k-anonymity of location
ϵ, δ	Privacy parameters
$\mathcal{K}$	Set of candidates for k
$\mathcal{M}_k$	Map of location anonymity sets for k
$\mathcal{A}_{PR}$	Adversary on producers and network routers
$\mathcal{A}_A$	Adversary on anonymizers

unique name called a *location name*, denoted by $n_i \in \mathcal{N}$, where $\mathcal{N}$ is the set of all location names for $\forall l_i \in \mathcal{L}$. The relationship between each location and its name is public information.

Content retrieval

Although IP is the universal network-layer protocol, current IP-based IoT solutions still face challenges; for example, IP addresses for a large number of IoT devices are depleted and additional name resolution mechanisms from an application-layer content name to the IP address of an IoT device are needed. To circumvent these problems, the research community has been exploring novel NDN-based IoT platforms [89]. As concrete instances, a keyword-based ICN-IoT platform [34] and name-based geographical forwarding [35, 36] have been proposed. An important feature of these platforms is that a consumer can retrieve location-related content of their LOI by sending an Interest packet specifying the location name assigned to the LOI. Specifically, a consumer retrieves a piece of location-related content by issuing a request specifying a pair of a location name, $n_i \in \mathcal{N}$, and a content type name,

$t_j \in \mathcal{T}$, where $\mathcal{T}$ is the set of possible content type names (e.g., $t_j = \texttt{temperature}$ or $\texttt{photo}$). A request is routed to l_i using n_i as a producer name (i.e., a routable prefix). Therefore, a request is not routed to l_i if n_i is encrypted. On receiving the request, a producer in l_i returns a reply containing the content payload corresponding to t_j and a signature σ_{sk_i}, where sk_i is the private key of the producer's public/private key pair (pk_i, sk_i). Integrity and provenance of the content can be verified using pk_i, and confidentiality of the content payload can be ensured in an end-to-end way by using some encryption scheme, such as attribute-based encryption, if required [77]. The request and reply are carried in an Interest packet and a Data packet, respectively. To deal with these platforms in a unified manner, the author represents them as $I(n_i, t_j)$ and $D(n_i, t_j, \sigma_{sk_i})$, respectively.

Next, the author models the choices of consumers for their LOIs. The author assumes that each consumer chooses a single location in $\mathcal{L}$ as the LOI of each request. Some previous studies assumed that the probability that each location is chosen as an LOI is only affected by the popularity of the location, which is defined as the ratio of the number of requests that specifies the location as LOIs to the total number of past requests of all consumers [40,63]. However, this is an unrealistic assumption because a consumer's choice of an LOI can also be affected by the consumer's preference and the LOIs of the consumer's past requests in general. For example, it is more likely that a consumer working at a university will choose the university as their LOI than other consumers. As another example, a consumer collecting video data along a road will choose a sequence of adjacent locations as their LOIs. From these observations, the author assumes that the choice of an LOI by consumer c depends on the preference of c in a snapshot request. In continual requests, the author assumes that an LOI can be determined depending on the LOIs of the past $r \in \mathbb{N}$ requests of c, denoted by $l_{i_1}, \cdots, l_{i_r}$, in addition to the preference of c. Consequently, the author assumes that a location l_i is chosen as an LOI according to the following probability:

$$\Pr[X_l = l_i \mid X_c = c, X_{l,-1} = l_{i_1}, \cdots, X_{l,-r} = l_{i_r}], \tag{4.1}$$

where X_l, $X_{l,-i}$, and X_c are random variables that describe an LOI of a request, the LOI of the request i times before the current request of a consumer, and a consumer.

Anonymizer

Anonymizers are used to provide consumer anonymity against an adversary on networks and generate location anonymity sets for consumers. The author assumes that anonymizers know for which service regions they generate location anonymity sets and the location names within them. Several anonymizers can offer their services for the same service region for load balancing. Because the anonymizers in this thesis are semi-honest and thus do not have to be completely trusted by all consumers, various candidates exist for the entity that provide the anonymizer service, such as ISPs, companies, and volunteers. First, as several existing anonymizing tools, any company can act as an anonymizer [54]. Second, several studies claim that ISPs should offer privacy services to their customers [90, 91]. Third, volunteers can act as anonymizers like anonymizing routers in Tor [18]. The author does not specify which of them manages anonymizers.

4.1.2 Adversary Model

The author assumes semi-honest adversaries who follow prescribed protocols but attempt to gain sensitive information regarding consumers from the protocols [92]. Specifically, the author considers the following two distinct semi-honest adversaries. First, the author assumes an adversary $\mathcal{A}_{PR}$ who compromises producers and network routers. This is equivalent to the adversary model in most previous studies. In addition, the author assumes the other adversary $\mathcal{A}_A$ who compromises an anonymizer. Note that the author assumes that $\mathcal{A}_{PR}$ and $\mathcal{A}_A$ do not collude with each other. The adversaries attempt to gain information from Interest/Data packets traversing them by leveraging their auxiliary information.

In terms of the auxiliary information of the adversaries, previous studies assume that each consumer chooses their LOIs depending only on the popularities of locations defined as probabilities $\Pr[X_l = l_i]$ $(1 \leq i \leq m)$ but not their preference and the LOIs of their past requests and then $\Pr[X_l = l_i]$ are the only auxiliary information of adversaries. In contrast, the author assumes that the adversaries are more powerful. First, the author assumes that the adversaries know the probability that a location anonymity set $\mathcal{S} \subseteq \mathcal{L}$ is chosen under the condition that l_i is chosen as the LOI (i.e., $\Pr[X_s = \mathcal{S} \mid X_l = l_i]$, where X_s is a random variable that describes a location anonymity set), in addition to $\Pr[X_l = l_i]$. This is possible because the author assumes that the algorithm for generating location anonymity sets, which determines this probability, is public. Second, the author assumes that the adversaries know the probability that l_i is chosen as an LOI by a particular consumer c who has chosen $l_{i_1}, \cdots, l_{i_r}$ as the LOIs of the past r requests (i.e., the probability represented by (4.1)).

4.1.3 Privacy Model

Section 4.1.3 explains how general privacy measures have been defined. Section 4.1.3 describes the requirements to achieve k-anonymity of location to prevent privacy violations caused by information leakage only from a location anonymity set. Section 4.1.3 shows the necessity to achieve consumer anonymity, defined as request indistinguishability, and k-anonymity of location simultaneously to protect location privacy against the adversaries leveraging their auxiliary information regarding requesting consumers.

Privacy Measures

Technical privacy measures are formalized using an adversary's prior belief and posterior belief regarding some sensitive information [93, 94]. Before an event occurs, an adversary has some belief regarding sensitive information of a consumer based on their auxiliary information. Such belief is called prior belief. An event can affect the prior belief and then the prior belief changes to posterior belief. The adversary infers the consumer's sensitive information by leveraging the posterior belief. Privacy measures are defined by how much information adversaries can obtain by leveraging posterior belief or the amount of revealed information, which is defined by the difference between prior belief and posterior belief.

K-anonymity of location

In terms of location privacy, sensitive information with respect to an LOI is classified as follows: (1) which location is an LOI; and (2) geographical information of an LOI. The most intuitive way for the adversaries to breach location privacy based on k-anonymity is to identify an LOI or exclude dummy location(s) from a location anonymity set; however, even if the adversaries cannot do so, they can still gain geographical information of an LOI. For example, if all the locations in a location anonymity set are contained in a small region, the adversaries can narrow down the LOI from the service region with some degree of accuracy. Leakage of this geographical information is unavoidable because a location anonymity set is a subset of the service region (i.e., $k < m$). Previous studies have proposed several requirements to prevent these two types of information leakage; however, they are only defined in ad-hoc manners to prevent specific attacks described in Section 4.3.3. The author defines requirements for a location anonymity set according to the rigorous definition of privacy by considering prior and posterior belief as follows:

- For locations in a location anonymity set, differences in posterior belief that each of the locations is the LOI are sufficiently small; and
- The difference between prior and posterior belief regarding the geographical information of an LOI caused by a choice of a location anonymity set is sufficiently small.

The first requirement implies that it is infeasible for the adversaries to infer the LOI from a location anonymity set. To this end, the author must ensure that all locations in a location anonymity set have sufficiently similar probabilities to be the LOI. To formalize this condition, the author uses entropy, which is widely used to measure equality of probability masses. Specifically, the author uses the entropy of X_l under the condition that X_s takes a location anonymity set $\mathcal{S} \subseteq \mathcal{L}$.

Definition 4.1.1 *Entropy of X_l under the condition that X_s takes $\mathcal{S}$ is defined as*

$$H(X_l \mid X_s = \mathcal{S}) = -\sum_{l_i \in s} \Pr[X_l = l_i \mid X_s = \mathcal{S}] \cdot \log_2(\Pr[X_l = l_i \mid X_s = \mathcal{S}]).$$

$\Pr[X_l = l_i \mid X_s = \mathcal{S}]$ is the probability that l_i is the LOI under the condition that $\mathcal{S}$ is chosen as the location anonymity set. From the definition of entropy, the more similar $\Pr[X_l = l_i \mid X_s = \mathcal{S}]$ ($l_i \in \mathcal{S}$) are, the larger $H(X_l \mid X_s = \mathcal{S})$ becomes. Therefore, the goal is to maximize $H(X_l \mid X_s = \mathcal{S})$. Specifically, let $num \in \mathbb{N}$ be the number of locations such that $\Pr[X_l = l_i \mid X_s = \mathcal{S}] > 0$ holds, $H(X_l \mid X_s = \mathcal{S})$ takes the optimal value $H(X_l \mid X_s = \mathcal{S}) = \log_2(num)$ when $\Pr[X_l = l_i \mid X_s = \mathcal{S}]$ take the same value for all locations in $\mathcal{S}$.

To formalize *num*, the author first defines $e_{i,\mathcal{S}}$ for each location $l_i \in \mathcal{L}$. Specifically, $e_{i,\mathcal{S}} = 1$ if $\Pr[X_l = l_i \mid X_s = \mathcal{S}] > 0$ and $e_{i,\mathcal{S}} = 0$ otherwise. By using $e_{i,\mathcal{S}}$, the author formalizes *num* using a function $size(\cdot)$ defined as follow:

Definition 4.1.2 *A location anonymity set $\mathcal{S} \subseteq \mathcal{L}$ is size k if it satisfies the following condition:*

$$size(\mathcal{S}) = |\{l_i \in \mathcal{S} \mid e_{i,\mathcal{S}} = 1\}| = k.$$

Intuitively, $size(\mathcal{S})$ represents the degree of k-anonymity of $\mathcal{S}$ because the set of locations that satisfy both $l_i \in \mathcal{S}$ and $\Pr[X_l = l_i \mid X_s = \mathcal{S}] > 0$ corresponds to the set of candidates of the LOI.

By using $H(X_l \mid X_s = \mathcal{S})$ and $size(\cdot)$, the first requirement can be defined as follows:

Definition 4.1.3 *Let $0 \leq \epsilon \leq 1$. A location anonymity set $\mathcal{S} \subseteq \mathcal{L}$ of size k is (k,ϵ)-secure if it satisfies the following condition:*

$$\frac{H(X_l \mid X_s = \mathcal{S})}{\log_2(size(\mathcal{S}))} \geq \epsilon.$$

The left side shows how close $H(X_l \mid X_s = \mathcal{S})$ is to the optimum value $\log_2(size(\mathcal{S}))$. Therefore, the sufficiently high ϵ helps prevent the adversaries from inferring the LOI in $\mathcal{S}$.

The second requirement is necessary to prevent the leakage of geographical information. Previous studies have attempted to achieve this goal by considering distances between k locations in a location

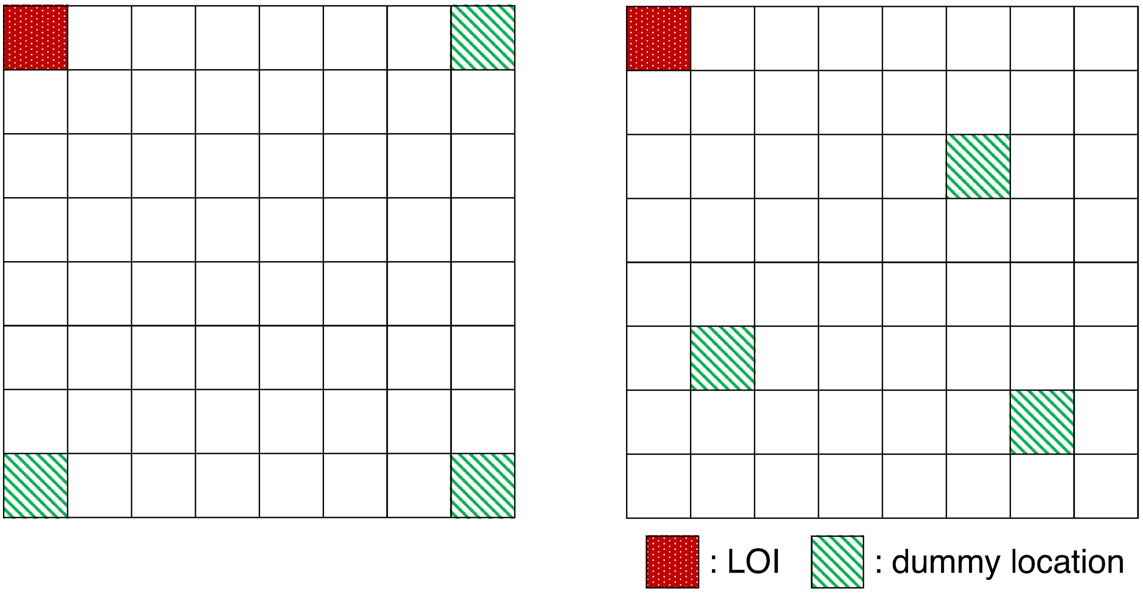


Figure 4.1: Comparison between examples of location anonymity sets generated according to the requirements in previous studies (left) and those in this thesis (right) ($k = 4$); each square is a location in $\mathcal{L}$.

anonymity set. However, these requirements are often inappropriate because locations that are as far apart as possible are chosen, such as the left illustration in Figure 4.1, and the adversaries can learn that the LOI of this request exists at the edge of the service region. In contrast, the goal is to generate each location anonymity set so that the locations are distributed uniformly throughout the service region including near the center of the service region, as shown in the illustration on the right of Figure 4.1, to minimize the geographical information that the adversaries can gain from it.

By applying the notion of t-closeness, the author defines a requirement for location anonymity sets to ensure that the difference between the geographical distribution of k locations in each location anonymity set and that of all the locations in the service region becomes sufficiently small. The author first defines a function $g(\cdot)$ that maps a set of locations to a probability distribution, representing the geographical distribution of a location anonymity set as follows:

Definition 4.1.4 *For a location anonymity set $\mathcal{S} \subseteq \mathcal{L}$, $g(\mathcal{S}) = (g_1, \cdots, g_m)$, where*

$$g_i = \begin{cases} \frac{1}{size(\mathcal{S})} & \textit{if } l_i \in \mathcal{S} \textit{ and } e_{i,\mathcal{S}} = 1 \\ 0 & \textit{otherwise}. \end{cases}$$

The author represents the coordinates of the location that is the x-th from the left and the y-th from the top of the two-dimensional service region as a tuple (x, y) and define the distance between two locations whose coordinates are (x_1, y_1) and (x_2, y_2) as $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ according to the 2D Euclidean distance. The distance between each pair of probability masses (g_i, g_j) in $g(\mathcal{S})$ is defined as the distance between the corresponding locations (l_i, l_j).

By definition, $g(\mathcal{S})$ represents the locations that are included in $\mathcal{S}$, the distances between pairs of locations, and the weight of each location in terms of geographical information leakage.

Next, the author defines a function $D(\cdot, \cdot)$ that maps a pair of probability distributions representing the geographical distributions of sets of locations to the distance between them. $D(\cdot, \cdot)$ is defined as the earth mover's distance (EMD) used in the t-closeness paper [41].

Definition 4.1.5 *Distance between two probability distributions representing geographical distributions of location anonymity sets $\mathcal{S}_i$ and $\mathcal{S}_j$ is defined as*

$$D(g(\mathcal{S}_i), g(\mathcal{S}_j)) = \mathsf{EMD}(g(\mathcal{S}_i), g(\mathcal{S}_j)),$$

where $\mathcal{S}_i, \mathcal{S}_j \subseteq \mathcal{L}$.

Specifically, the EMD is formulated as a transportation problem and measures the least amount of work required to turn a probability distribution into the other by transferring probability masses according to the size of each probability mass and the distances between two probability masses [95]. Because EMD is a distance function that considers the distances between probability masses, the closer the locations in two sets are, the smaller the EMD becomes. The author defines $\mathcal{L}_0$ as the set of all locations in $\mathcal{L}$ that cannot be chosen as LOIs of consumers (i.e., $\mathcal{L}_0 = |\{l_i \in \mathcal{L} \mid \Pr[X_l = l_i] = 0\}|$). Thus, with respect to the fixed value of $g(\mathcal{L} \setminus \mathcal{L}_0)$, the more the locations in a location anonymity set $\mathcal{S}$ are distributed uniformly throughout the service region, the smaller $\mathsf{EMD}(g(\mathcal{S}), g(\mathcal{L} \setminus \mathcal{L}_0))$ becomes.

By using $g(\cdot)$ and $D(\cdot, \cdot)$, the second requirement can be defined as follows:

Definition 4.1.6 *Let $\delta \geq 0$. A location anonymity set $\mathcal{S} \subseteq \mathcal{L}$ of size k is (k,δ)-distributed if it satisfies the following condition:*

$$D(g(\mathcal{S}), g(\mathcal{L} \setminus \mathcal{L}_0)) \leq \delta.$$

The sufficiently low δ helps prevent the adversaries from narrowing a possible region including the LOI to a small region because locations in a location anonymity set are distributed uniformly throughout the service region.

Combining the features of a (k,ϵ)-secure and a (k,δ)-distributed location anonymity set, the author defines a (k,ϵ,δ)-private location anonymity set as follows:

Definition 4.1.7 *A location anonymity set $\mathcal{S} \subseteq \mathcal{L}$ of size k is (k,ϵ,δ)-private if it is both (k,ϵ)-secure and (k,δ)-distributed.*

Consumer anonymity

In the previous subsection, it is assumed that the adversaries do not leverage auxiliary information on a requesting consumer. Because the adversaries observe Interest/Data packets, they can gain information not only regarding which location anonymity set is specified in a request but also regarding which consumer sends the request and regarding which location anonymity sets have been specified in the past requests made by the consumer. Specifically, the adversaries can infer an LOI of a request using the following probabilities, which can be derived from the auxiliary information represented by (4.1):

$$\Pr[X_l = l_i \mid X_c = c, X_s = s, X_{s,-1} = \mathcal{S}_{i_1}, \cdots, X_{s,-r} = \mathcal{S}_{i_r}], \tag{4.2}$$

where X_s and $X_{s,-1}$ are random variables that describe a location anonymity set specified in the current request and that specified in the request i times before, respectively. For example, in the case in which a consumer who works at a university chooses a location containing the university as an LOI, the adversaries can correctly infer the LOI with a high probability, even if the LOI cannot be inferred only from the corresponding location anonymity set.

To prevent such an attack, each location anonymity set must be generated so that the entropy derived by replacing $\Pr[X_l = l_i \mid X_s = \mathcal{S}]$ in Definition 4.1.1 with the probability represented in (4.2) is sufficiently large. Such a location anonymity set generation process is said to be Bayes optimal in terms of generating the best location anonymity set considering the auxiliary information of the adversaries. However, it is difficult to achieve Bayes optimal privacy in reality, as discussed by Machanavajjahala et al. [96]. One of the most crucial factors is that the auxiliary information of the adversaries is not known for other entities in general. In addition, even a consumer would not know the probability distribution based on which LOI is chosen in each request. For example, a consumer who works at a university would not know how likely it is that a location in the university will be chosen as LOIs of their future requests.

Therefore, the author takes an approach that prevents the adversaries from using the auxiliary information on consumers by achieving consumer anonymity, defined as request indistinguishability, instead of aiming to generate location anonymity sets that achieve Bayes optimal privacy. Specifically, the author defines consumer anonymity as follows:

Definition 4.1.8 *A consumer who issues requests is said to be anonymous if the adversaries who observes the requests cannot identify the consumer and distinguish whether the requests are sent by the same consumer or by different consumers.*

If consumer anonymity is achieved, the adversaries can infer an LOI only by leveraging a location anonymity set specified in a request because they cannot identify the consumer who sent the request and which location anonymity sets have been specified in the requests before the request.

4.2 Private Location-related Content Retrieval in NDN

The previous section showed that it is necessary to achieve both k-anonymity of location and consumer anonymity against $\mathcal{A}_{PR}$ and $\mathcal{A}_A$ to protect location privacy. This section describes a system to achieve this goal, called PLCR.

4.2.1 Overview

In most previous studies, a trusted anonymizer receives a consumer's LOI, generates a location anonymity sets for the LOI, and sends back the piece of content corresponding to the LOI to the consumer; however, this approach is vulnerable to the semi-honest anonymizers (i.e., $\mathcal{A}_A$). To achieve k-anonymity of location against $\mathcal{A}_A$, the author leverages maps of location anonymity sets and a PIR-based scheme. Figure 4.2 illustrates the overview of PLCR.

An anonymizer generates maps of location anonymity sets for varying $k \in \mathcal{K}$, where $\mathcal{K}$ is the pre-defined set of all possible degree of k-anonymity of location. Specifically, each of the maps contains a set of location anonymity sets and their identifiers. Consumers learn which location anonymity set should be used when each of the location in the service region is chosen as the LOI from a map. Suppose a consumer has one of the maps of location anonymity sets, denoted by $\mathcal{M}_k$. First, a consumer and an anonymizer exchange a secret key used for encrypting their communication. To achieve consumer anonymity against $\mathcal{A}$, this key exchange is performed so that consumer anonymity is achieved against an anonymizer. Then, the consumer sends a request specifying the identifier of a

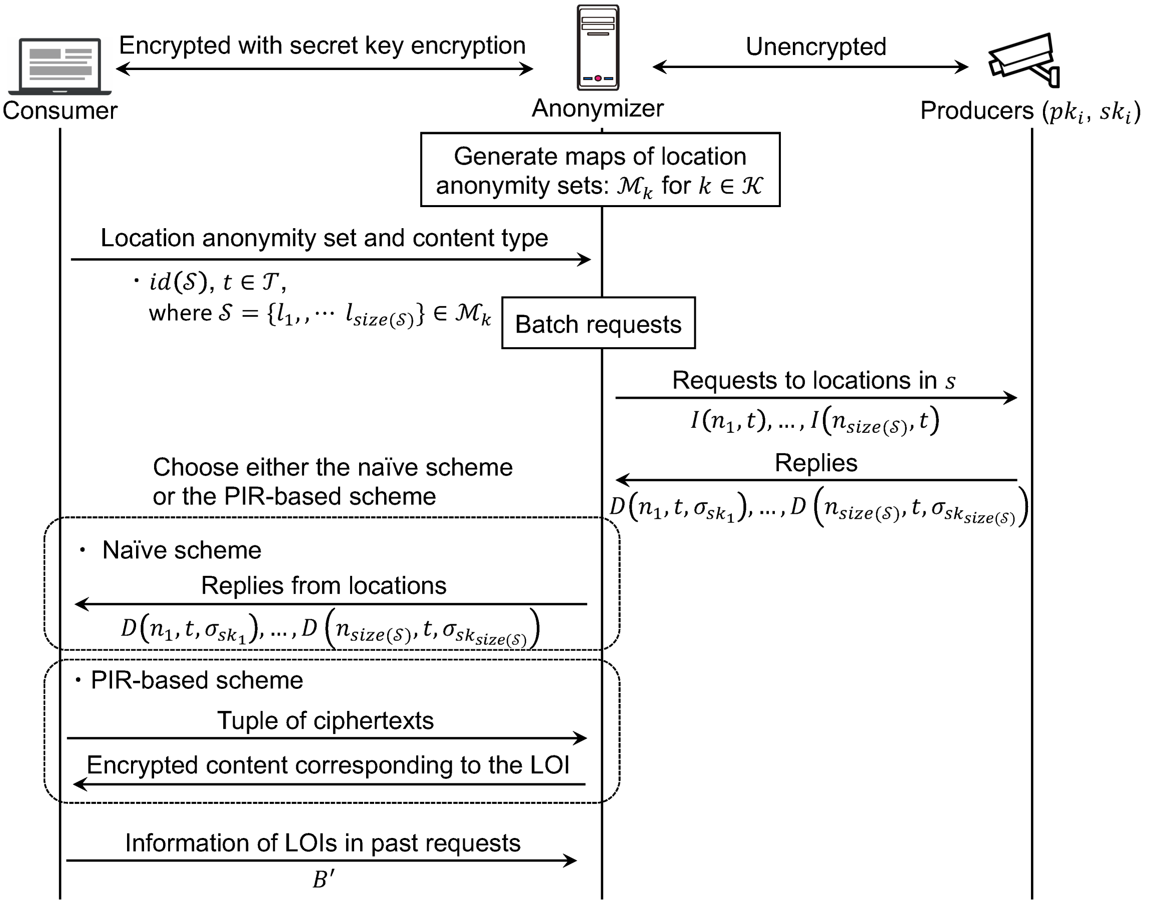

Encrypted with secret key encryption
Unencrypted
Consumer
Anonymizer
Producers (pk_i, sk_i)
Generate maps of location anonymity sets: $\mathcal{M}_k$ for $k \in \mathcal{K}$
Location anonymity set and content type
· $id(\mathcal{S})$, $t \in \mathcal{T}$,
where $\mathcal{S} = \{l_1, , \cdots l_{size(\mathcal{S})}\} \in \mathcal{M}_k$
Batch requests
Requests to locations in s
$I(n_1, t), \dots, I(n_{size(\mathcal{S})}, t)$
Replies
$D(n_1, t, \sigma_{sk_1}), \dots, D\left(n_{size(\mathcal{S})}, t, \sigma_{sk_{size(\mathcal{S})}}\right)$
Choose either the naïve scheme
or the PIR-based scheme
· Naïve scheme
Replies from locations
$D(n_1, t, \sigma_{sk_1}), \dots, D\left(n_{size(\mathcal{S})}, t, \sigma_{sk_{size(\mathcal{S})}}\right)$
· PIR-based scheme
Tuple of ciphertexts
Encrypted content corresponding to the LOI
Information of LOIs in past requests
B'

Figure 4.2: Overview of PLCR.

location anonymity sets of their choice $\mathcal{S} = \{l_1, \cdots, l_{size(\mathcal{S})}\} \in \mathcal{M}_k$ and a content type $t \in \mathcal{T}$ to the anonymizer, instead of the LOI itself. Section 4.2.2 describes how an anonymizer generates the maps.

On receiving the request, the anonymizer retrieves content from all the locations in the location anonymity set by sending a set of Interest packets $I(n_1, t), \cdots, I(n_{size(\mathcal{S})}, t)$, where each n_i is the location name of $l_i \in \mathcal{S}$. Therefore, k-anonymity of location is achieved against $\mathcal{A}_{PR}$ in the same way as the previous studies. At some point, the corresponding Data packets $D(n_1, t, \sigma_{sk_1}), \cdots, D(n_{size(\mathcal{S})}, t, \sigma_{sk_k})$ are returned to the anonymizer. The naïve scheme to retrieve content corresponding to the LOI while hiding the LOI from the anonymizer is that the consumer receives all the Data packets from the anonymizer. However, its crucial disadvantage is that a consumer must receive all the pieces of content returned from $l_1, \cdots, l_{size(\mathcal{S})}$. Section 4.2.3 shows that a PIR-based scheme is promising to enable a consumer to receive only the content corresponding to their LOI while reducing the amount of data to receive but requiring some additional computation. Consumers can choose either the naïve scheme or the PIR-based scheme depending on their communication and computation capabilities.

In addition, consumer anonymity against $\mathcal{A}_{PR}$ is achieved by an anonymizer acting like a mix router, which forwards the Interest/Data packets from consumers in a random order after batching them for a short time [97], as described in Section 4.2.4. Therefore, $\mathcal{A}_{PR}$ cannot correlate incoming and outgoing packets at an anonymizer and thus $\mathcal{A}_{PR}$ cannot identify which location anonymity set is specified by a consumer. Moreover, the author carefully designs PLCR so that a consumer can anonymously communicate with an anonymizer by leveraging the feature of NDN that Interest packets do not carry any information regarding their senders to achieve consumer anonymity against $\mathcal{A}_A$.

Finally, consumers periodically upload statistics on the LOIs of their past requests, denoted by B', to the anonymizer while hiding the exact LOIs by leveraging a scheme based on locally differentially private (LDP) protocols. The anonymizer estimates the popularities of locations by aggregating the statistics from consumers and updates the maps of location anonymity set using the popularities if necessary. Section 4.2.2 describes how to derive B'.

4.2.2 Location Anonymity Set Generation and Retrieval

Because $\Pr[X_l = l_i \mid X_s = \mathcal{S}] = \Pr[X_s = \mathcal{S} \mid X_l = l_i]\Pr[X_l = l_i]/\Pr[X_s = \mathcal{S}]$ holds according to Bayes' theorem, an anonymizer must obtain an estimate of the popularities of locations $p = (p_1, \cdots, p_m)$, where p_i is an estimate of $\Pr[X_l = l_i]$, to maximize $H(X_l \mid X_s = \mathcal{S})$ of each location anonymity set $\mathcal{S}$. To this end, an anonymizer needs to collect information on which location consumers have chosen as their LOIs; however, this contradicts the purpose of protecting location privacy against $\mathcal{A}_A$. Therefore, the author needs a technique that enables an anonymizer to obtain the statistics regarding which location consumers have chosen as more frequently as their LOIs and derive p, while hiding the exact LOI of each request. Some previous studies have proposed a method in which each consumer passes information on his/her past LOIs to access points while walking around locations and collects information on other consumers' LOIs from the access points at the same time to estimate the popularities of locations [40, 63]. However, this method has drawbacks in that the LOIs of consumers are revealed to the possibly compromised access points and that they need to walk around until they exchange information with many access points.

LDP protocols have been developed to enable a data collector to gather statistics of users while preserving their privacy [44]. As a concrete example, Google Chrome leverages an LDP protocol to gather users' answers regarding their browsing settings, such as their default homepages and search engines [98]. The author describes how to derive p based on the concept of LDP protocols.

A consumer who has sent a request creates a tuple, $B = (b_1, \cdots, b_m)$, where $b_i = 1$ if l_i is chosen as the LOI of the request and $b_i = 0$ otherwise. Then, the consumer randomizes B so that the LOI is not identified from B. Concretely, the consumer generates another tuple $B' = (b'_1, \cdots, b'_m)$ according to the following probability:

$$\Pr[b'_i = 1] = \begin{cases} q & \text{if } b_i = 1 \\ 1 - q & \text{otherwise;} \, and \end{cases}$$

$$\Pr[b'_i = 0] = \begin{cases} q & \text{if } b_i = 0 \\ 1 - q & \text{otherwise,} \end{cases}$$

where q is a constant such that $0 < q < 1$ and $q \neq 1/2$. Such a tuple B' is referred to as a *feedback*. Consumers can upload such feedbacks anonymously by sending each feedback in an Interest packet since Interest packets carry no information on their senders.

Suppose an anonymizer has received $f \in \mathbb{N}$ feedbacks from consumers and f_i is the number of times l_i has actually been chosen as the LOIs of the corresponding requests. Let the number of elements in B' with $b'_i = 1$ be denoted by h_i; then, $\mathsf{Exp}[h_i] = q \cdot f_i + (1 - q) \cdot (f - f_i)$ holds. Thus, $\hat{f}_i = \frac{h_i - (1-q) \cdot f}{2q - 1}$ is an unbiased estimate of f_i. By using $\hat{f}_i$ $(1 \leq i \leq m)$, the anonymizer derives $p_i = \frac{\hat{f}_i}{\sum_{j=1}^{m} \hat{f}_j}$.

It has been pointed out that this feedback scheme is vulnerable in scenarios where an adversary can correlate multiple feedbacks from a specific consumer. To solve this problem, several privacy-enhancing techniques have been proposed [98]; however, such techniques are not required in PLCR for the following two reasons: (1) each consumer uploads the feedback for each request only once; and (2) the anonymizer cannot correlate multiple feedbacks from a specific consumer with each other because each consumer anonymously uploads them to the anonymizer. Consequently, each feedback is equivalent to a one-time collection from the perspective of the anonymizer and thus it is sufficient to focus only on the privacy leakage from each feedback.

After deriving p, an anonymizer generates location anonymity sets using the algorithm described in Section 4.3.2. Because consumers may require various levels of location privacy [99], an anonymizer generates location anonymity sets for all $k \in \mathcal{K}$ to enable each consumer to choose a degree of k-anonymity for their requests. For each $k \in \mathcal{K}$, an anonymizer generates location anonymity sets for all possible LOIs and publishes the map of location anonymity set, denoted by $\mathcal{M}_k$, indicating which location anonymity set a consumer should specify when each location is chosen as an LOI. Specifically, an anonymizer generates location anonymity sets so that the service region is divided disjointly, i.e., no location is included in more than one location anonymity set, to achieve large ϵ, as described in Sec 4.3.1. Thus, a consumer holding one of the maps can uniquely choose the location anonymity set for their LOI.

The author assumes that an anonymizer assigns an unique identifier for each of the location anonymity sets (e.g., distinct integers). The identifier of a location anonymity set $\mathcal{S}$ is denoted by $id(\mathcal{S})$. Note that the author assumes that the maps are not tampered by an anonymizer because they are semi-honest. Because p can change over time, an anonymizer periodically verifies whether every location anonymity set contained in the maps has sufficiently high ϵ and low δ; if not, the anonymizer must regenerate the maps. The author assumes that the expiration times of the maps are explicitly stated and if a consumer does not have a map or if their map has expired, the consumer chooses $k \in \mathcal{K}$ and retrieves the corresponding map from an anonymizer.

Note that the anonymizer needs to initialize the popularities of locations with some realistic values if a sufficient number of feedbacks have not been returned from consumers. To this end, the anonymizer can leverage the number of IoT devices (i.e., producers) installed in each location, assuming that locations with more IoT devices are more likely to be chosen as LOIs. Specifically, in a scenario where vehicles act as producers of location-related content, as the author assumes in Section 4.4.1, existing spatial point processes [100] that model the random locations of vehicles in cities can be leveraged to derive the number of producers.

4.2.3 Achieving K-anonymity of Location

The overview of a PIR protocol is explained in Section 2.2.3. Among several concrete protocols to realize PIR, homomorphic public key encryption-based schemes are promising. In general, with these schemes, a user first generates a public/private key pair and creates a tuple of size u, where u is the size of a database. The i-th item in the tuple is 1 and the others are 0, where i is the index of the item the user is interested in. Next, the user encrypts each item with the public key using a homomorphic public key encryption algorithm and send the tuple of ciphertexts to a database. The database multiplies each pair of items in the tuple and the database with the same index, adds the resulting ciphertexts all together, and sends the ciphertext back to the user. Thanks to the homomorphic property, the items in the database that are multiplied by encrypted 0 are ignored when they are summed up, and only the item that is multiplied by encrypted 1 is extracted. Finally, the user obtains the i-th item by decrypting the ciphertext with the private key.

By regarding an anonymizer that has retrieved k pieces of content as the database server in the PIR-based schemes (i.e., $u = k$), a consumer can retrieve content corresponding to their LOI while achieving k-anonymity of location against the anonymizer. Suppose a consumer has a public/private key pair (pk, sk) and a map $\mathcal{M}_k$. The consumer first chooses their LOI and obtains a location anonymity set $\mathcal{S} \in \mathcal{M}_k$ which contains the LOI. After that, the consumer sends an anonymizer a request specifying $id(\mathcal{S})$ and a tuple of ciphertexts. The anonymizer sends Interest packets to all the locations in $\mathcal{S}$ and receive corresponding Data packets. For the received tuple of ciphertexts and Data packets, the anonymizer creates the reply for the consumer. Finally, the consumer can obtain the piece of content corresponding the LOI by decrypting the reply.

Because the reply from an anonymizer contains only one ciphertext, the PIR-based scheme potentially reduces the communication cost between a consumer and an anonymizer. The author compares the naïve scheme and the PIR-based scheme in terms of their communication and computation cost in Section 4.4.2 and show that the PIR-based scheme actually reduces the communication cost with a reasonable computation cost. Although communication between an anonymizer and producers still requires exchange of at least k pairs of Interest/Data packets in plaintext, this communication overhead can be reduced by content caches of network routers and an anonymizer.

4.2.4 Achieving Consumer Anonymity

Network routers forward packets without using source and destination addresses, thereby naturally achieving consumer anonymity against $\mathcal{A}_A$. However, $\mathcal{A}_{PR}$ can break consumer anonymity by obtaining the MAC address of a consumer at their first-hop network router if the consumer communicate with an anonymizer in plaintext. To prevent this attack, the overall communication between a consumer and an anonymizer must be encrypted with a CCA-secure encryption scheme. An important requirement for the encryption scheme is that an anonymizer must not gain any information regarding the identity of a consumer. Therefore, the author employs a key exchange protocol for NDN that exchanges a secret key without revealing the identity of a consumer, such as CCNxKE [81]. The author also assumes that each consumer changes a secret key with an anonymizer for each request.

Next, $\mathcal{A}_{PR}$ can launch the following attacks to break consumer anonymity: (1) content correlation and (2) timing correlation [47]. With both attacks, $\mathcal{A}_{PR}$ attempt to correlate incoming packets at an anonymizer to the corresponding outgoing packets.

First, content correlation is an attack that correlates incoming and outgoing packets by leveraging their content and sizes. PLCR prevents this attack by secret key encryption/decryption at an anonymizer and packet padding. Obviously, encryption/decryption at an anonymizer prevents $\mathcal{A}_{PR}$ from correlating packets with their content. In addition, a consumer and an anonymizer append padding texts to names and payloads of Interest/Data packets so that the sizes of all packets are the same.

Second, PLCR prevents timing correlation, which correlates incoming and outgoing packets by leveraging the times of reception and transmission, by accumulating $batch \in \mathbb{N}$ requests and the corresponding replies. In general, the larger *batch* is, the stronger consumer anonymity is achieved because each consumer is hidden in more consumers. This type of packet scheduling scheme is similar to the previous studies on mix routers [47, 97]. To suppress the delay, an anonymizer can generate decoy Interest packets by randomly choosing location anonymity sets from the map $\mathcal{M}_k$ if *batch* Interest packets do not arrive at an anonymizer within a predefined time period $delay \in \mathbb{R}$. This method is vulnerable to a typical attack against mix routers that an adversary injects Interest packets that will fill the anonymizer's buffer for *batch* Interest packets to degrade consumer anonymity. However, such an attack can be mitigated by having the anonymizer send decoy Interest packets even if it has received *batch* Interest packets from consumers, as proposed in [47]. The author evaluates the probability that an anonymizer generates decoy interest packets in Section 4.4.3.

4.3 Location Anonymity Set Generation

This section describes the design rationale for an algorithm for generating location anonymity set as an optimization problem, present the concrete algorithm, and analyze its security against attacks considered in previous studies.

4.3.1 Optimization Problem Formulation

Owing to a constraint of PLCR, an anonymizer generates location anonymity sets corresponding to all possible LOIs, as described in Section 4.2. Thus, the two parameters ϵ and δ are defined for each location anonymity set. Let ϵ_i and δ_i be the values of ϵ and δ of the i-th location anonymity set, respectively. Because even one location anonymity set with small ϵ_i or large δ_i causes location privacy leakage for consumers who use it, the objectives for generating location anonymity sets are to maximize the minimum value of ϵ_i and to minimize the maximum value of δ_i simultaneously.

$$\text{maximize: } \min_i \epsilon_i \tag{4.3}$$

$$\text{minimize: } \max_i \delta_i \tag{4.4}$$

To achieve (4.4), location anonymity sets must be generated so that they contain locations that are distributed uniformly throughout the service region. To achieve (4.3), the following three design aspects must be considered: (1) geographical constraints, (2) similarity in popularities of locations (3) disjoint location anonymity sets.

First, each location anonymity set must have a size of k or more, i.e., it must consist of more than k locations that can be chosen as an LOI, for each $k \in \mathcal{K}$. If a location anonymity set contains a location that is never chosen as an LOI owing to geographical constraints, for example, a location in the sea, the adversaries can immediately eliminate the location from the LOI candidates. The author assumes that the probability $\Pr[X_l = l_i]$ for every location is marked by an anonymizer in advance, and such locations are not assigned to any location anonymity set.

The second and third ones are necessary to prevent the adversaries from inferring an LOI by keeping $H(X_l \mid X_s = \mathcal{S})$ high to maximize ϵ_i. To this end, each location anonymity set $\mathcal{S}$ must contain locations that have similar values of $\Pr[X_l = l_i \mid X_s = \mathcal{S}]$. According to the Bayes' theorem,

$$\Pr[X_l = l_i \mid X_s = \mathcal{S}] = \frac{\Pr[X_s = \mathcal{S} \mid X_l = l_i]\Pr[X_l = l_i]}{\sum_{l \in \mathcal{S}} \Pr[X_s = \mathcal{S} \mid X_l = l]\Pr[X_l = l]}, \tag{4.5}$$

holds, where $\Pr[X_s = \mathcal{S} \mid X_l = l_i]$ represents the probability that $\mathcal{S}$ is chosen under the condition that l_i is an LOI. Therefore, $\Pr[X_l = l_i]$ and $\Pr[X_s = \mathcal{S} \mid X_l = l_i]$ must be considered simultaneously. The previous studies have attempted to achieve this goal by choosing k locations that have similar values of $\Pr[X_l = l_i]$ [40,63]; however, this corresponds to only the second design aspect. In addition, the author generates location anonymity sets so that they are disjoint with respect to each other. Consequently, $\Pr[X_s = \mathcal{S} \mid X_l = l_i] = 1$ holds for $\forall l_i \in \mathcal{S}$, and thus,

$$\Pr[X_l = l_i \mid X_s = \mathcal{S}] = \frac{\Pr[X_l = l_i]}{\sum_{l \in \mathcal{S}} \Pr[X_l = l]}, \tag{4.6}$$

holds. Therefore, the first objective (4.3) is achieved by generating all location anonymity sets so that they contain locations that have similar values of popularities $\Pr[X_l = l_i]$.

From the first and the third design aspect, the following constraints for the optimization problem are derived, where $\mathcal{S}_i, \mathcal{S}_j \subseteq \mathcal{L}$ are two distinct location anonymity sets.

$$\text{subject to:} \quad \forall i, size(\mathcal{S}_i) \geq k \tag{4.7}$$

$$\bigcup_i \mathcal{S}_i = \mathcal{L} \setminus \mathcal{L}_0 \tag{4.8}$$

$$\forall i, j \; (i \neq j), \mathcal{S}_i \cap \mathcal{S}_j = \emptyset \tag{4.9}$$

4.3.2 Heuristic Algorithm

The author designs an algorithm for generating location anonymity sets according to the optimization problem. Although the algorithm has two objectives and three constraints, the objectives cannot be simultaneously optimized in general. This is because they are affected by the popularities of locations and the ways in which they are distributed in the service region. For example, k locations with the most similar popularities are not always distributed throughout the service region. Because optimal k-anonymization is an NP-hard problem as shown in [101], the author aims to design a heuristic algorithm, hereinafter. Concretely, the author adopts the approach of modifying a heuristic algorithm originally proposed for k-anonymization of databases to generate location anonymity sets. This approach has advantages in that their performance and security have been well studied and that better algorithms for k-anonymization of databases that will emerge in the future can also be applied to protect location privacy. This approach becomes possible because the author defined requirements for k-anonymity of the location following the strict definitions of privacy refined for database k-anonymization.

The author leverages the Mondrian algorithm [45], which is an efficient greedy algorithm to divide entries of a database into anonymity sets satisfying k-anonymity with time complexity $O(n \log n)$, where n is the number of entries. After sorting entries focusing on an attribute of the quasi-identifiers of the database, the Mondrian algorithm recursively divides the entries into two partitions using the median of the attribute as a division point. Before each division, it checks whether the resulting two partitions contain at least k entries, and if both satisfy this condition, they are left as candidates for further division. This procedure is repeated until there are no more divisible partitions. Consequently, k-anonymized partitions, each of which contains entries whose quasi-identifier values are as similar as possible, are generated.

An overview of the proposed Mondrian-based algorithm is summarized in Algorithm 1. In the same way as the original Mondrian algorithm, the algorithm has time complexity $O(m \log m)$. Suppose the information of the candidates of LOIs in the service region (i.e., the set of locations l_i

Algorithm 1: Location anonymity set generation

```
Input: $p$: Estimate of the popularities of locations
       $k$: Degree of $k$-anonymity of location
       $\delta$: Metric for $(k,\delta)$-location anonymity set
Output: $P_1, \cdots, P_n$: Set of partitions, each of which
        corresponds to a location anonymity sets
Function Anonymize(P, $P_0$):
    median ← The median of entry.popularity for entry $\in P$
    $P_1 \leftarrow \{entry \in P \mid entry.popularity \leq$ median$\}$
    $P_2 \leftarrow \{entry \in P \mid entry.popularity >$ median$\}$
    if $size(P_1) < k$ or $size(P_2) < k$ then
        return $P$
    else if $D(g(P_1), g(P_0)) \leq \delta$ or $D(g(P_2), g(P_0)) \leq \delta$ then
        return $P$
    else
        return Anonymize($P_1$, $P_0$), Anonymize($P_2$, $P_0$)
Function Main:
    $P_0 \leftarrow \{\}$
    foreach location $l_i \in \mathcal{L} \setminus \mathcal{L}_0$ do
        $P_0$.append($\{p_i$, the coordinates of $l_i\}$)
    return Anonymize($P_0$, $P_0$)
```

that have $\Pr[X_l = l_i] > 0$) is summarized as entries of a database whose attributes are the estimate of the popularity p_i (represented by *entry.popularity*) and the coordinates of each location. The author generates location anonymity sets, each of which contains locations with similar p_i, and thus, achieve large ϵ_i values, with the Mondrian algorithm using the popularity attribute and the coordinate attribute as a quasi-identifier and a sensitive-attribute, respectively. Moreover, the author extends the algorithm so that it checks not only the k-anonymity requirement but also whether the two divided partitions have sufficiently small values of δ_i by calculating $D(\cdot,\cdot)$ between locations in each partition and those in the overall entries to achieve small δ_i values.

This algorithm satisfies the three constraints. First, (4.7) is satisfied because the partition corresponding to a single location anonymity set is divided under the constraint that it contains at least k entries in each division procedure. Second, (4.8) and (4.9) are satisfied because each entry corresponds to one location without overlapping, and disjoint partitions are derived from all entries.

4.3.3 Security Analysis

Previous studies have proposed several attacks to violate k-anonymity of location by exploiting the vulnerabilities of location anonymity set generation algorithms and defenses against the attacks. However, the defenses are designed in ad-hoc manners, and the conditions for defending all attacks have not been thoroughly discussed. This subsection shows that the algorithm is resilient to such attacks.

Center of location anonymity set attack

Kalnis et al. indicated that several algorithms generate a location anonymity set whose LOI tends to be its center because they simply choose locations around an input LOI as dummy locations [37]. If such algorithms are used, locations near the center of a location anonymity set are highly likely to be an LOI. In contrast, the algorithm generates location anonymity sets for all possible LOIs so that the k locations in each location anonymity set are equally likely to be an LOI and a consumer chooses one of them according to their LOI, thereby preventing this attack.

Intersection attack

Several deterministic algorithms that output a location anonymity set for an inputted LOI are vulnerable to a intersection attack [38]. With this attack, adversaries attempt to identify the LOI of a location anonymity set specified in a request by inputting all the locations in it to the algorithm as an LOI and compare the outputs. This attack is successful if there exists a location $l_i \in \mathcal{S}$ such that the algorithm outputs different location anonymity sets $\mathcal{S}' \neq \mathcal{S}$. If such a location exists, adversaries can eliminate l_i from the candidates of an LOI. The algorithm is deterministic but is resilient to the intersection attack because it divides the service region into disjoint-location anonymity sets without taking an LOI as the input.

Inference attack

With an inference attack, adversaries attempt to infer an LOI from a location anonymity set $\mathcal{S}_j$ by leveraging the probabilities, $\Pr[X_l = l_i \mid X_s = \mathcal{S}_j]$ for all $l_i \in \mathcal{S}_j$. Existing algorithms defend this attack by choosing dummy locations so that they have similar popularities ($\Pr[X_l = l_i]$) with an LOI as much as possible [40, 102]; however, this approach is not sufficient. As discussed in Section 4.1.3, probabilities $\Pr[X_s = \mathcal{S}_j \mid X_l = l_i]$, which depend on the algorithm, must be considered in addition to $\Pr[X_l = l_i]$. The algorithm is resilient to the inference attack because the author designed the algorithm so that $\Pr[X_s = \mathcal{S}_j \mid X_l = l_i] = 1$ holds for $\forall l_i \in \mathcal{S}_j$ by generating disjoint location anonymity sets. Thus, the entropy of each location anonymity set is maximized by combining locations with similar values of $\Pr[X_l = l_i]$.

Location similarity attack

Previous studies indicated that adversaries can narrow down a potential region in which an LOI is included if all the k locations in a location anonymity set are within a small region [39,40,64]. To solve this problem, several studies have proposed requirements for location anonymity sets; however, it has not been completely solved, as explained in Section 4.1.3. The algorithm generates a location anonymity set so that the potential regions become as large as possible by adapting the concept of t-closeness proposed for privacy in the database to protect location privacy.

4.4 Evaluation

This section evaluates the quality of location anonymity sets generated by the algorithm proposed in Section 4.3.2, the overhead incurred by using a PIR-based protocol to achieve k-anonymity of location against $\mathcal{A}_A$, the probability that an anonymizer generates decoy packets to achieve consumer anonymity against $\mathcal{A}_{PR}$, and the size of $\mathcal{M}_k$.

4.4.1 Location Anonymity Set

Simulation condition

The author assumes vehicles in an urban area as IoT devices which provide consumers with collected data such as their speed values and video data of their surroundings as location-related content. Since the proposed algorithm aims to generate each location anonymity set by combining locations that have similar popularities *and* are far apart from each other, it can fail to generate good location anonymity sets if all locations with similar popularities are within a narrow range. To avoid using such an unsuitable service region, a large service region including a sufficiently large number of locations should be used. Taking this feature into account, to derive a probability distribution representing the popularities of locations (i.e., $\Pr[X_l = l_i]$), the author conducted a simulation of vehicle movements as follows: the author used the simulation of urban mobility (SUMO) simulator [103] to simulate vehicle movements according to the measurement results in Luxembourg [104], which is a well-known realistic scenario. The size of the service region, i.e., the center of Luxembourg, is approximately 60 km^2 and it is divided into 1024 locations. The author simulated the vehicle movements in the service region for 24 hours and measured the number of vehicles in each location for each 30 minutes time period. The author derived the popularity of l_i as the number of vehicles in l_i normalized by the total number of vehicles in the service region, assuming that the number of times each location is chosen as LOIs within each time period is proportional to the number of vehicles in the location. The author set $\Pr[X_l = l_i] = 0$ for location l_i where vehicles cannot obviously exist owing to its geographical constraints.

To evaluate the proposed algorithm for various distributions of popularities of locations, the author generates location anonymity sets by the proposed algorithm according to the popularities of locations for every time period and evaluates their ϵ_i and δ_i values. As inputs to the algorithm, the author chose $k = 10$ and $\delta = 7.0$. In addition, the author compares the location anonymity sets with those generated with another algorithm designed according to the same optimization problem [105]. The overview of the existing algorithm is summarized as follows: The service region is divided into k segments, each of which has at least $\lfloor size(\mathcal{L} \setminus \mathcal{L}_0)/k \rfloor$ locations satisfying $\Pr[X_l = l_i] = 1$, along the domain of definition of Z-order [106], which is a coordinate system based on a space filling curve. Each location anonymity set is generated by choosing one location from each of the segments, assuming that two locations that are far apart from each other on the Z-order are actually far apart in terms of the 2D Euclidean distance. To maximize $H(X_l \mid X_s = \mathcal{S})$ of each location anonymity set $\mathcal{S}$, the locations are chosen in descending order of their popularities. By iterating this procedure until all locations are assigned to location anonymity sets, the service region is naturally divided disjointly. Note that the author does not conduct comparison with the algorithms proposed in other studies because their requirements for location anonymity sets are substantially different from those defined by the author and thus they cannot achieve k-anonymity of location, as shown in Section 4.3.3. These algorithms are implemented in Python and use an efficient EMD calculation algorithm proposed in [107, 108].

(k,ϵ)-secure location anonymity set

Figure 4.3 shows the time series of the maximum, median, and minimum ϵ_i values of generated location anonymity sets. To understand the causes of fluctuation in ϵ_i, the author also evaluated the variance of the number of vehicles at each location. Figure 4.3 and Figure 4.4 show the same results, whereas Figure 4.4 shows ϵ_i in a narrower range [0.94, 1.0]. The following two observations are obtained: First, the minimum values of ϵ_i are smaller than the optimum value 1.0 for all time intervals. Nevertheless, the author concludes that the proposed generation algorithm generates sufficiently good

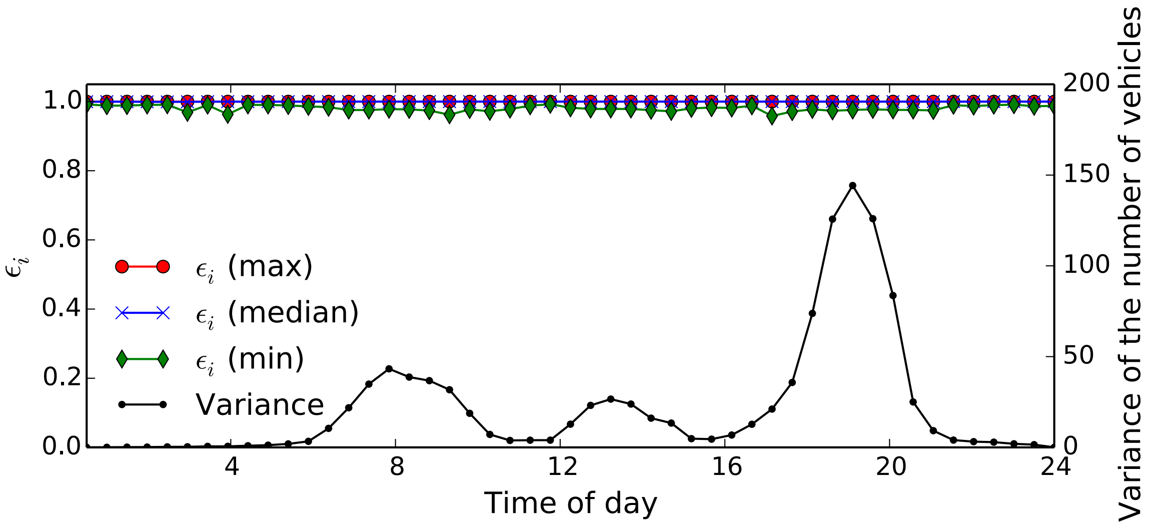


Figure 4.3: Time series of the maximum, the median, and the minimum values of ϵ_i.

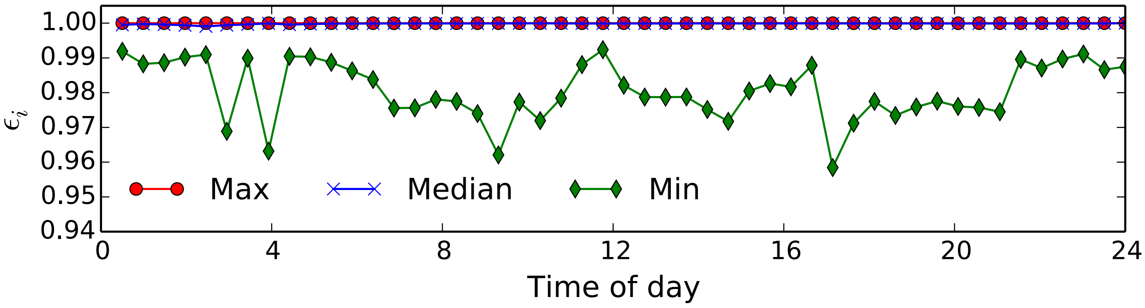


Figure 4.4: Time series of the maximum, the median, and the minimum values of ϵ_i shown in a narrower range.

location anonymity sets because the median values of ϵ_i are almost the same as the optimum values for all time intervals. Second, the minimum values of ϵ_i decreases as the variance of the number of vehicles increases. The high variance of the number of vehicles is the same as the high variance of the popularities of locations and thus it is intrinsically difficult to keep ϵ_i of all the generated location anonymity sets high by choosing k locations with similar popularity values in such cases.

Figure 4.5 shows the time series of the minimum values of ϵ_i of location anonymity sets generated by the algorithm in this thesis and that in the existing study. The author concludes that the algorithm in this thesis provides stronger k-anonymity of location because it achieves better results for all time intervals. The reason is that the previous algorithm divides the service region into multiple segments and only one location in each segment is chosen as a candidate for a location anonymity set, while the

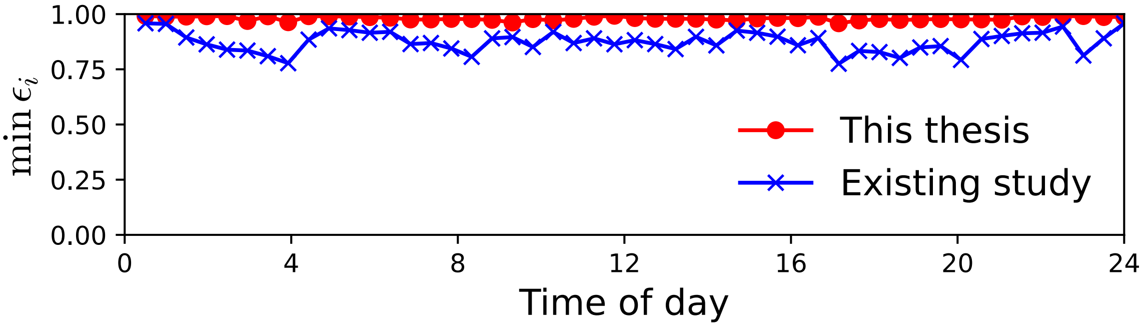


Figure 4.5: Time series of the minimum values of ϵ_i of location anonymity sets generated by the algorithm proposed in this thesis and that in the existing study.

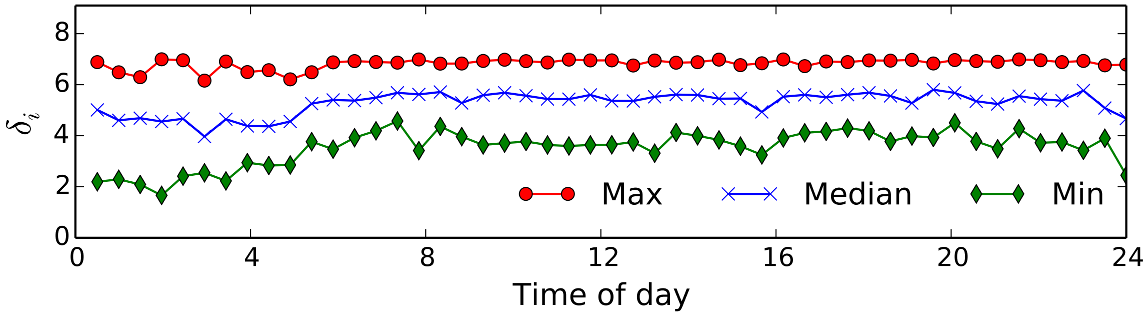


Figure 4.6: Time series of the maximum, the median, and the minimum values of δ_i.

algorithm in this thesis chooses locations with similar popularities from all locations in the service region.

(k,δ)-distributed location anonymity set

Next, the author evaluates the values of δ_i to confirm that their locations are well distributed uniformly throughout the service region. Figure 4.6 shows the maximum, the mean, and the minimum values of δ_i in the time series of location anonymity set generation. From the results, the author concludes that the generated location anonymity sets have reasonable values of δ_i and thus it is difficult for the adversaries to narrow down the range in which an LOI exists in an unseemly small range.

Figure 4.7 shows the time series of the maximum values of δ_i of location anonymity sets generated by the algorithm in this thesis and that in the existing study. The former and the latter algorithm have similar values on the average, but the former is more stable. This is because the latter uses Z-order to measure the distance between two locations. Specifically, the latter divides the service region into segments according to z-order to choose the locations distributed throughout the service

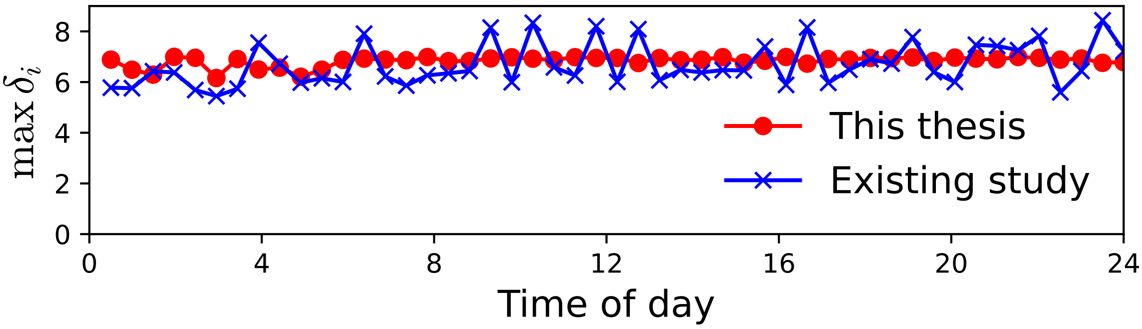


Figure 4.7: Time series of the maximum values of δ_i of location anonymity sets generated by the algorithm proposed in this thesis and that in the existing study.

region. However, two locations are not always far apart from each other in terms of the 2D Euclidean distance even if they are far apart in terms of Z-order. In contrast, the former checks δ values multiple times when dividing service region into partitions to decide whether each partition should be further divided and thus stable δ_i values can be obtained.

To show that locations in each location anonymity set generated with the algorithm are well distributed uniformly throughout the service region, the author presents examples of location anonymity sets whose δ_i correspond to the median and the max value in a time period in Figure 4.8. Gray-colored squares indicate the locations that can be chosen as LOIs (i.e., locations such that $\Pr[X_l = l_i] > 0$ hold), and blue-colored squares indicate the locations included in the location anonymity sets. The location anonymity sets include 11 and 13 locations ($> k = 10$), respectively. From Figure 4.8, the author concludes that even locations in the worst location anonymity set (i.e., the location anonymity set that has the largest δ_i) are distributed sufficiently uniformly throughout the service region and thus prevents the adversaries to obtain geographical information of LOIs.

4.4.2 Overhead to Achieve K-anonymity of Location

To compare the communication and the computation cost in the naïve scheme and those in the PIR-based scheme described in Section 4.2.3, the author implemented the prototypes of these schemes as applications that run on a consumer and an anonymizer. To implement the PIR-based scheme, the author uses SealPIR [43], which is one of the most efficient PIR schemes based on a homomorphic public key encryption scheme. The rationale for choosing SealPIR is that it is efficient especially for consumers. In general, PIR schemes have been designed to enable a user to retrieve the i-th item of a remote database with u items. SealPIR is efficient for users because it requires each query to contain

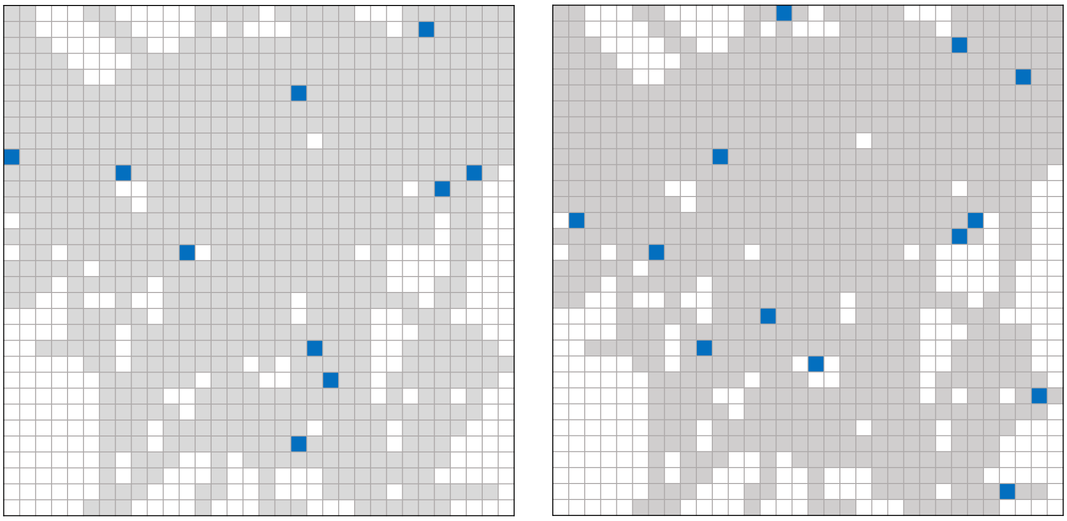


Figure 4.8: Location anonymity sets whose δ_i correspond to the median (left) and the max value (right) in a time period.

d ciphertext for some constant $d \in \mathbb{N}$ ($d < u$), while other homomorphic public key encryption scheme-based PIR schemes like XPIR [109] requires exactly u ciphertexts. A drawback of SealPIR is that it requires additional computations on the database server to expand the d ciphertexts to u ciphertexts; however, because consumers typically have lower resources compared to anonymizers, efficiency for consumers is more preferred.

In PLCR, a consumer uses the PIR-based scheme to retrieve the piece of content corresponding to their LOI from k pieces of content. Therefore, u in SealPIR corresponds to k in PLCR and a consumer and an anonymizer play the roles of a user and a database server, respectively. To incorporate SealPIR into PLCR, the author performed a parameter optimization. With SealPIR, the collection of items is represented as a d-dimensional hypercube and the number of items u must be no more than κ^d, where κ is the security parameter for the underlying public key encryption scheme (typically, $\kappa = 2048$). Therefore, large d is required for large u. However, large d makes the number of ciphertexts in each query. Fortunately, in the setting of this thesis, consumers set k regardless of the number of locations m and typically $k \ll \kappa$. Therefore, $d = 1$ is sufficient for reasonable values of k and this enables each query of consumers to contain only one ciphertext.

The author evaluates the process delay at a consumer and an anonymizer and the size of data that a consumer receives from an anonymizer. For the PIR-based scheme, the process delay at a consumer is defined as the total time it takes to generate a query and extract the plaintext Data packet corresponding to the LOI from the ciphertexts received from an anonymizer, and the process delay

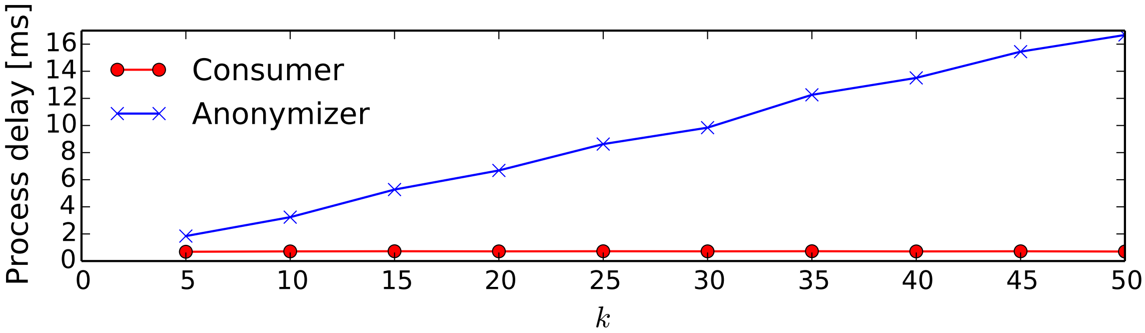


Figure 4.9: Process delay at a consumer and an anonymizer for varying k (data size is 1 KB).

at the anonymizer is defined as the total time it takes to derive a response to a consumer from the received Data packets. Because the naïve scheme does not require computationally heavy operations such as public key encryption, the author does not evaluate the process delay for the naïve scheme. For both the naïve scheme and the PIR-based scheme, the size of data that a consumer receives from an anonymizer is defined as the total size of ciphertexts the consumer receives for each query. Note that the experiment here does not consider the delay due to the packet batching at an anonymizer. The author sets the size of each ciphertext to 32KB following the default setting of SealPIR. The applications on consumers and anonymizers are implemented in C++ by using SealPIR [43] and OpenSSL. All experiments were conducted on a machine with an Intel(R) Xeon(R) E5-2620 v4 processor (2.10 GHz) with eight DDR4 16GB DRAM devices, running Ubuntu 18.04 LTS. The author used AES-256 as the secret key encryption/decryption scheme for the naïve scheme.

Figure 4.9 shows the process delay as a function of the anonymity degree $k \in \mathcal{K} = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$ (the size of each Data packet is 1 KB). Similarly, Figure 4.10 shows the process delay as a function of the size of each Data packet ($k = 20$). It is shown that the overhead due to the process delay is sufficiently small for both consumers and anonymizers. In particular, owing to the feature of SealPIR described above, the process delay at a consumer is almost constant whereas the process delay at an anonymizer increases as k or the data size increases. Therefore, a consumer who prefers strong location privacy protection can choose a large value of k without worrying about their computation capability. However, this increases the communication costs between an anonymizer and producers and thus an anonymizer should appropriately set $\mathcal{K}$.

Figure 4.11 shows the sizes of data as a function of the anonymity degree k. With the naïve scheme, naturally, the size scales linearly with k. In contrast, with the PIR-based scheme, the size is 32 KB regardless of k for $k \leq \kappa$ (i.e., a consumer receives only one ciphertext). Since the size of ciphertext is set to 32KB, each consumer always receives 32KB data to retrieve content of size less

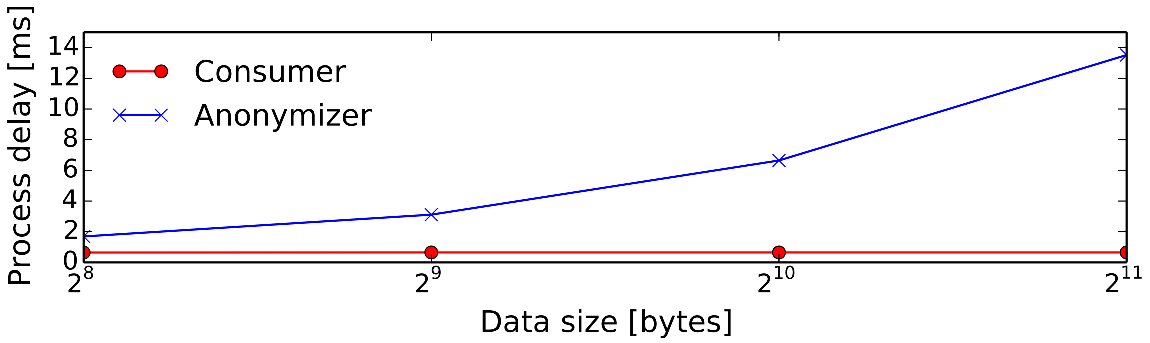


Figure 4.10: Process delay at a consumer and an anonymizer for varying data size ($k = 20$).

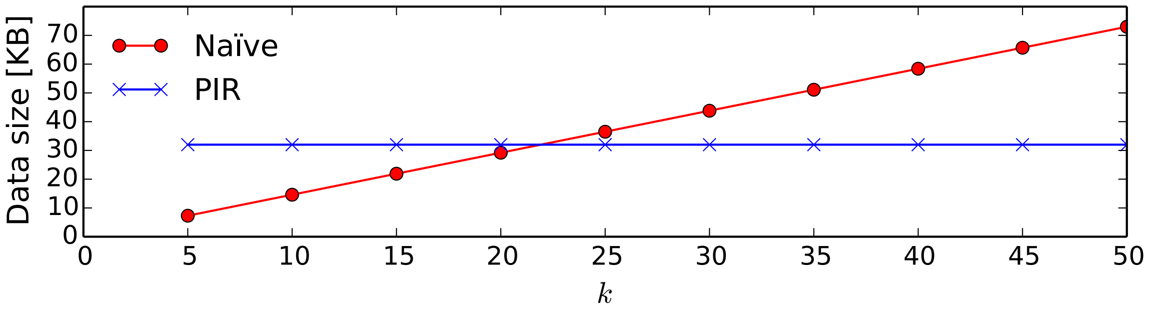


Figure 4.11: Size of data that a consumer receives from an anonymizer.

than 32KB in the current implementation of SealPIR. Conversely, if consumers can retrieve content of size more than 32KB, it is necessary to set the size of each ciphertext to a larger value (e.g., 64KB) in advance. However, this increases the network costs. From the result, the author concludes that a consumer who has a reasonable computation capability and prefers a large value of k can leverage the PIR-based scheme to reduce the size of data to receive.

4.4.3 Decoy Packets Insertion

As described in Section 4.2.4, an anonymizer accumulates *batch* inputted Interest and Data packets to achieve consumer anonymity. If *batch* Interest packets do not arrive at an anonymizer in a predefined time period *delay*, it generates random decoy Interest packets. Therefore, the maximum delay incurred by an anonymizer is *delay*, and this delay is the influential factor of the delay at an anonymizer. Thus, the author evaluates the probability of decoy Interest packets being inserted under the following assumptions: Interest packets arrive at an anonymizer according to the Poisson process with a rate λ, which is defined as the number of Interest packets arriving at the anonymizer per second, and producers immediately send back Data packets after receiving Interest packets. Therefore, the

Table 4.2: The Probability of Decoy Packets Being Inserted.

batch \ λ	5	10	15	20
5	0.44	0.03	0.00	0.00
10	0.97	0.46	0.07	0.01
15	0.99	0.92	0.47	0.11

probability of decoy packets being inserted depends on the Interest packet arrival rate λ.

Under these assumptions, the probability that $\mu \in \mathbb{N}$ Interest packets arrive at an anonymizer within the time period *delay* is derived as $p(\mu, delay, \lambda) = \frac{e^{-\lambda \cdot delay}(\lambda \cdot delay)^{\mu}}{\mu!}$.

Let decoy be an event in which at least one decoy Interest packet is inserted, and $\Pr[\mathsf{decoy}]$ be the probability that this event occurs. $\Pr[\mathsf{decoy}]$ is equivalent to the probability that *batch* Interest packets do not arrive at an anonymizer within *delay* and thus it is derived as follows:

$$\Pr[\mathsf{decoy}] = \sum_{\mu=0}^{batch-1} p(\mu, delay, \lambda). \tag{4.10}$$

The author sets *delay* to 1.0 [s] because *delay* must be smaller than the Pending Interest Table (PIT) expiration time in NDN, which is determined based on network RTTs, to transport Data packets to consumers. The author analyzes $\Pr[\mathsf{decoy}]$ in the case where *batch* is set to 5, 10, and 15, and λ is set to 5, 10, 15, 20, respectively. The results are depicted in Table 4.2. $\Pr[\mathsf{decoy}]$ decreases as *batch* decreases or λ increases, and decoy Interest packets are, therefore, rarely inserted in the cases in which *batch* is sufficiently small (i.e., each consumer enjoys relatively weak anonymity) or λ is sufficiently high (i.e., there are a lot of requesting consumers). Consequently, an anonymizer should insert many decoys Interest packets if strong consumer anonymity must be achieved against $\mathcal{A}_{PR}$ but there are only a few consumers.

4.4.4 Size of a Map of Location Anonymity Sets

The semi-honest anonymizer in this thesis sends a map of location anonymity sets $\mathcal{M}_k$ to consumers, whereas the honest anonymizer in the existing studies needs not do so. Thus, the size of $\mathcal{M}_k$ should be small.

In the design of PLCR, an anonymizer generates disjoint location anonymity sets. Thus, the only information that $\mathcal{M}_k$ should contain is which location anonymity set each location is included in and this can be represented by an integer. Therefore, the size of the map $\mathcal{M}_k$ is derived by $m\times$ (the size of an integer). In the case $m = 1024$, as in the simulation in Section 4.4.1, the size of an integer is 4 bytes long, and the size of $\mathcal{M}_k$ is just 4 KB. This is sufficiently small for consumers to retrieve periodically.

4.5 Discussion and Conclusion

This chapter rigorously defined the requirements for a location anonymity set to achieve k-anonymity of location based on the privacy model elaborated in database field and designed an algorithm for generating location anonymity set. In addition, the author showed that consumer anonymity must be provided together with k-anonymity of location to protect location privacy under the case where consumers continually retrieve content from their LOIs. To this end, the author designed PLCR to achieve k-anonymity of location and consumer anonymity simultaneously, assuming that the anonymizer is semi-honest.

The author's future research plans include designing a location privacy protection system that uses some techniques other than k-anonymity. An inherent disadvantage of k-anonymity-based location privacy protection systems like PLCR is that an anonymizer sends at least k requests to producers simultaneously and thus the communication cost increases in proportion to k. In-network caching of NDN potentially reduces this cost; however, it is still unacceptable for a centralized producers accommodating large number of consumers. A differential privacy-based system that conceals an LOI by adding random noise to the location information (e.g., the x-coordinate and y-coordinate of an LOI) is as promising as PLCR. Although such a system does not increase the communication cost, location privacy is easily violated if a consumer continually sends requests, owing to the limitation of differential privacy. Combining differential privacy with consumer anonymity to guarantee that consumers' continual requests are not correlated to each other can circumvent the above disadvantage of differential privacy. In addition, a comprehensive comparison of k-anonymity-based and differential privacy-based location privacy is another promising research direction.

Chapter 5

Conclusion

In this thesis, the author studied anonymity and privacy in NDN. By treating content as the first-class entity, NDN yields several advantages for achieving efficient content distribution; however, human-readable content names and publicly verifiable signatures make sufficient levels of anonymity and privacy difficult to provide in NDN. The importance of anonymity and privacy on the network has been emphasized in recent years and thus the author believes that the above feature is one of the factors hindering the popularization of NDN. To solve this problem, the author designed two systems that provide sufficient levels of anonymity and privacy by compensating the above weakness and taking advantage of a strength of NDN that Interest packets do not carry any information regarding their senders. The author comprehensively analyzes the levels of anonymity and privacy the proposed systems provide through theoretical and/or empirical ways. The author believes that these results contribute to making NDN a serious candidate for the next-generation network architecture.

In the first part of this thesis, the author studied producer anonymity. My main focus has been on a formal definition and an onion routing-based system to provide producer anonymity. First, the author revealed that producer anonymity on the content-oriented NDN network should be defined differently from the conventional anonymity definitions for the host-oriented IP network. The author defined producer anonymity by focusing on packet flow on the network to capture the session-less feature of NDN. Next, the author designed ACPNDN as the first system to provide producer anonymity under a realistic adversary model. The author carefully designed ACPNDN to take advantage of NDN that Interest packets do not carry any information regarding their senders by leveraging the RICE protocol. The author analyzed the level of anonymity provided by ACPNDN according to the definition of producer anonymity, showing that ACPNDN reduces RTT for content retrieval compared to hidden service. In conjunction with existing studies on consumer anonymity, this study gives a fundamental solution to anonymity on NDN networks. Moreover, the author argues that the design rationale of ACPNDN is not unique to content-centric architectures like NDN; it can be applied to host-centric architectures that conceal source addresses or routes to sources with encryption like APIP [110], LAP [111], PHI [112]. In these architectures, a consumer builds a path to a producer in advance to forward packets according to routing information concealed with encryption. To achieve producer anonymity, the identity of a producer should be concealed from the consumer and network routers; however, such techniques have not been fully discussed. This study can contribute to designing anonymity systems for such new host-centric architectures.

In the second part of this thesis, the author studied k-anonymity-based location privacy. First, the author rigorously defined the requirements to prevent adversaries from inferring the LOI from a location anonymity set and to minimize the leakage of geographical information of an LOI by using the notions of entropy and t-closeness, respectively. The empirical analysis showed that the location anonymity set generation algorithm proposed in this study provides sufficiently secure location anonymity sets. Second, the author then designs a system for private retrieval of location-related content, called PLCR, assuming that the anonymizer is a semi-honest adversary. This new but realistic adversary model causes the following two problems that have not been addressed in the existing studies: the communication cost increases linearly with k and the anonymizer cannot compute the popularities of locations, which are required to generate secure location anonymity sets. To address these problems, PLCR leverages PIR and LDP protocols. These protocols allow each consumer to retrieve the piece of content corresponding to their LOIs by encrypting the LOI with a homomorphic encryption scheme and to periodically upload statistics on the LOIs of their past requests to the anonymizer while hiding the exact LOIs, respectively. The author empirically showed that PLCR is especially suitable for a consumer who has a reasonable computation capability and prefers a large value of k. Moreover, consumer anonymity provided at the network layer contributes to protecting location privacy for multiple requests by a consumer. The location privacy protection technique proposed in this study can be applied to many location-based services (e.g., map services, navigation services, and location-based games/SNS), regardless of the network architecture.

Bibliography

[1] T. Anderson, K. Birman, R. Broberg, M. Caesar, D. Comer, C. Cotton, M. J. Freedman, A. Haeberlen, Z. G. Ives, A. Krishnamurthy, *et al.*, “The nebula future internet architecture,” in *The Future Internet Assembly*, pp. 16–26, Springer, 2013.

[2] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri, “Mobilityfirst future internet architecture project,” in *Proceedings of the 7th Asian Internet Engineering Conference*, pp. 1–3, 2011.

[3] D. Han, A. Anand, F. Dogar, B. Li, H. Lim, M. Machado, A. Mukundan, W. Wu, A. Akella, D. G. Andersen, *et al.*, “{XIA}: Efficient support for evolvable internetworking,” in *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pp. 309–322, 2012.

[4] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, kc claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, “Named data networking,” *ACM SIGCOMM Computer Communication Review*, vol. 44, pp. 66–73, July 2014.

[5] S. Farrell and H.Tschofenig, “Pervasive Monitoring Is an Attack,” RFC 7258, RFC Editor, May 2014.

[6] J. Griffiths, “China is exporting the great firewall as internet freedom declines around the world,” 2018.

[7] G. T. Aliza Vigderman, “The data big companies have on you,” 2022.

[8] P. Gasti and G. Tsudik, “Content-centric and named-data networking security: The good, the bad and the rest,” in *2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, pp. 1–6, June 2018.

[9] M. Kim, J. Lim, H. Yu, K. Kim, Y. Kim, and S.-B. Lee, “Viewmap: Sharing private in-vehicle dashcam videos,” in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, (Boston, MA), pp. 163–176, USENIX Association, Mar. 2017.

[10] K. Kita, Y. Kurihara, Y. Koizumi, and T. Hasegawa, “Location privacy protection with a semi-honest anonymizer in information centric networking,” in *Proceedings of ACM Conference on Information-Centric Networking*, 2018.

[11] I. Miers, C. Garman, M. Green, and A. D. Rubin, “Zerocoin: Anonymous distributed e-cash from bitcoin,” in *2013 IEEE Symposium on Security and Privacy*, pp. 397–411, May 2013.

[12] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, “Mixcoin: Anonymity for bitcoin with accountable mixes,” in *Financial Cryptography and Data Security*, (Berlin, Heidelberg), pp. 486–504, Springer Berlin Heidelberg, 2014.

[13] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, *Freenet: A Distributed Anonymous Information Storage and Retrieval System*, pp. 46–66. Springer Berlin Heidelberg, 2001.

[14] R. Dingledine, M. J. Freedman, and D. Molnar, *The Free Haven Project: Distributed Anonymous Storage Service*, pp. 67–95. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.

[15] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger, “Infranet: Circumventing web censorship and surveillance,” in *11th USENIX Security Symposium (USENIX Security 02)*, 2002.

[16] S. DiBenedetto, P. Gasti, G. Tsudik, and E. Uzun, “ANDāNA: Anonymous named data networking application,” *ArXiv e-prints*, Dec. 2011.

[17] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, “Anonymous connections and onion routing,” *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 482–494, May 1998.

[18] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: The second-generation onion router,” in *Proceedings of the 13th conference on USENIX Security Symposium*, pp. 1–17, Aug. 2004.

[19] A. Pfitzmann and M. Hansen, “Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management - consolidated proposal for terminology,” *Version v0*, vol. 31, 01 2007.

[20] C. Díaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Privacy Enhancing Technologies* (R. Dingledine and P. Syverson, eds.), (Berlin, Heidelberg), pp. 54–68, Springer Berlin Heidelberg, 2003.

[21] P. Zhang, Q. Li, and P. P. C. Lee, "Achieving content-oriented anonymity with crisp," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, pp. 578–590, Nov 2017.

[22] J. Feigenbaum, A. Johnson, and P. Syverson, "A model of onion routing with provable anonymity," in *Financial Cryptography and Data Security*, (Berlin, Heidelberg), pp. 57–71, Springer Berlin Heidelberg, 2007.

[23] The Tor Project, "Tor rendezvous specification - version 3," 2017.

[24] M. Krol, K. Habak, D. Oran, D. Kutsher, and P. Ioannis, "Rice: Remote method invocation in icn," in *Proceedings of ACM Conference on Information-Centric Networking*, 2018.

[25] L. Overlier and P. Syverson, "Locating hidden servers," in *2006 IEEE Symposium on Security and Privacy (S P'06)*, pp. 15 pp.–114, May 2006.

[26] M. Wright, M. Adler, B. N. Levine, and C. Shields, "Defending anonymous communications against passive logging attacks," in *2003 Symposium on Security and Privacy, 2003.*, pp. 28–41, May 2003.

[27] Foursquare, "Foursquare," 2017.

[28] M. Gruteser and B. Hoh, "On the anonymity of periodic location samples," in *International Conference on Security in Pervasive Computing*, pp. 179–192, Springer, 2005.

[29] C. Bettini, X. S. Wang, and S. Jajodia, "Protecting privacy against location-based personal identification," in *Workshop on Secure Data Management*, pp. 185–199, Springer, 2005.

[30] J. R. Vacca. Boston: Morgan Kaufmann, third edition ed., 2017.

[31] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, p. 31–42, 2003.

[32] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of*

the 2013 ACM SIGSAC Conference on Computer & Communications Security, p. 901–914, 2013.

[33] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: Anonymizers are not necessary," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, p. 121–132, 2008.

[34] O. Ascigil, S. Reñé, G. Xylomenos, I. Psaras, and G. Pavlou, "A keyword-based icn-iot platform," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, p. 22–28, 2017.

[35] R. Kai, K. Yuki, and T. Hasegawa, "Name-based geographical routing/forwarding support for location-based iot services," in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pp. 1–6, 2016.

[36] N. Yang, K. Chen, and Y. Liu, "Towards efficient ndn framework for connected vehicle applications," *IEEE Access*, vol. 8, pp. 60850–60866, 2020.

[37] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1719–1733, 2007.

[38] N. Talukder and S. I. Ahamed, "Preventing multi-query attack in location-based services," in *Proceedings of the Third ACM Conference on Wireless Network Security*, p. 25–36, 2010.

[39] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: Query processing for location services without compromising privacy," in *Proceedings of the 32nd International Conference on Very Large Data Bases*, p. 763–774, VLDB Endowment, 2006.

[40] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *Proceedings of IEEE INFOCOM*, pp. 754–762, 2014.

[41] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Proceedings of IEEE International Conference on Data Engineering*, pp. 106–115, 2007.

[42] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: single database, computationally-private information retrieval," in *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pp. 364–373, 1997.

[43] S. Angel, H. Chen, K. Laine, and S. Setty, “Pir with compressed queries and amortized query processing,” in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 962–979, 2018.

[44] S. L. Warner, “Randomized response: A survey technique for eliminating evasive answer bias,” *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[45] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, “Mondrian multidimensional k-anonymity,” in *22nd International conference on data engineering (ICDE’06)*, pp. 25–25, IEEE, 2006.

[46] D. L. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” *Commun. ACM*, vol. 24, p. 84–90, Feb. 1981.

[47] C. Gulcu and G. Tsudik, “Mixing e-mail with babel,” in *Proceedings of Internet Society Symposium on Network and Distributed Systems Security*, pp. 2–16, Feb 1996.

[48] A. I. P. P. T. M. P. L. S. N. A. S. Ulf Moeller, Lance Cottrell, “Mixmaster Protocol Version 2,” Internet-Draft draft-sassaman-mixmaster-03, Internet Engineering Task Force, Dec. 2004. Work in Progress.

[49] G. Danezis, R. Dingledine, and N. Mathewson, “Mixminion: Design of a type iii anonymous remailer protocol,” in *2003 Symposium on Security and Privacy, 2003.*, pp. 2–15, IEEE, 2003.

[50] S. J. Murdoch and G. Danezis, “Low-cost traffic analysis of tor,” in *2005 IEEE Symposium on Security and Privacy (S P’05)*, pp. 183–195, May 2005.

[51] S. Chakravarty, A. Stavrou, and A. D. Keromytis, “Identifying proxy nodes in a tor anonymization circuit,” in *2008 IEEE International Conference on Signal Image Technology and Internet Based Systems*, pp. 633–639, Nov 2008.

[52] P. Mittal, A. Khurshid, J. Juen, M. Caesar, and N. Borisov, “Stealthy traffic analysis of low-latency anonymous communication using throughput fingerprinting,” in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS ’11, (New York, NY, USA), pp. 215–226, ACM, 2011.

[53] V. Shmatikov and M.-H. Wang, “Timing analysis in low-latency mix networks: Attacks and defenses,” in *Computer Security – ESORICS 2006*, (Berlin, Heidelberg), pp. 18–33, Springer Berlin Heidelberg, 2006.

[54] Anonymizer, “Anonymizer,” 1995.

[55] R. Tourani, S. Misra, J. Kliewer, S. Ortegel, and T. Mick, "Catch me if you can: A practical framework to evade censorship in information-centric networks," in *Proceedings of the 2Nd ACM Conference on Information-Centric Networking*, ACM-ICN '15, (New York, NY, USA), pp. 167–176, ACM, 2015.

[56] J. Kurihara, K. Yokota, and A. Tagami, "A consumer-driven access control approach to censorship circumvention in content-centric networking," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, ACM-ICN '16, (New York, NY, USA), pp. 186–194, ACM, 2016.

[57] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for web transactions," *ACM Trans. Inf. Syst. Secur.*, vol. 1, p. 66–92, Nov. 1998.

[58] D. Chaum, "The dining cryptographers problem: Unconditional sender and recipient untraceability," *Journal of cryptology*, vol. 1, no. 1, pp. 65–75, 1988.

[59] D. Chaum and E. van Heyst, "Group signatures," in *Proceedings of Eurocrypt 1991, volume 547 of LNCS, pages 257–65.*, p. 257–265, 1991.

[60] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '01, (Berlin, Heidelberg), pp. 552–565, Springer-Verlag, 2001.

[61] S. Kaushik, R. Tourani, G. Torres, S. Misra, and A. Afanasyev, "Ndn-abs: Attribute-based signature scheme for named data networking," in *Proceedings of the ACM Conference on Information-Centric Networking*, ACM-ICN '19, (New York, NY, USA), pp. 123–133, 2019.

[62] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[63] B. Niu, Q. Li, X. Zhu, and H. Li, "A fine-grained spatial cloaking scheme for privacy-aware users in location-based services," in *Proceedings of International Conference on Computer Communication and Networks*, pp. 1–8, IEEE, 2014.

[64] H. Lu, C. S. Jensen, and M. L. Yiu, "Pad: Privacy-area aware, dummy-based location privacy in mobile services," in *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, p. 16–23, 2008.

[65] R.-H. Hwang, Y.-L. Hsueh, and H.-W. Chung, “A novel time-obfuscated algorithm for trajectory privacy protection,” *IEEE Transactions on Services Computing*, vol. 7, no. 2, pp. 126–139, 2013.

[66] Y. Wang, J. Peng, L.-p. He, T.-t. Zhang, and H.-z. Li, “Lbss privacy preserving for continuous query based on semi-honest third parties,” in *2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC)*, pp. 384–391, 2012.

[67] C. Dwork, “Differential privacy: A survey of results,” in *Theory and Applications of Models of Computation* (M. Agrawal, D. Du, Z. Duan, and A. Li, eds.), (Berlin, Heidelberg), pp. 1–19, Springer Berlin Heidelberg, 2008.

[68] A. Khoshgozaran, C. Shahabi, and H. Shirani-Mehr, “Location privacy: going beyond k-anonymity, cloaking and anonymizers,” *Knowledge and Information Systems*, vol. 26, no. 3, pp. 435–465, 2011.

[69] G. Tsudik, E. Uzun, and C. A. Wood, “AC^{3}N: An api and service for anonymous communication in content-centric networking,” in *Proceedings of IEEE Annual Consumer Communications Networking Conference*, pp. 988–991, Jan. 2016.

[70] T. Dierks and E. Rescorla, “The transport layer security (tls) protocol version 1.2,” RFC 5246, August 2008.

[71] E. Rescorla, “The transport layer security (tls) protocol version 1.3,” RFC 8446, August 2018.

[72] J. Iyengar and M. Thomson, “QUIC: A UDP-Based Multiplexed and Secure Transport,” Tech. Rep. 9000, May 2021.

[73] J. Katz and Y. Lindell, “Introduction to modern cryptography,” 2007.

[74] C. Ghali, G. Tsudik, and C. A. Wood, “(the futility of) data privacy in content-centric networking,” in *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, WPES ’16, (New York, NY, USA), p. 143–152, ACM, 2016.

[75] X. Cui, Y. H. Tsang, L. C. K. Hui, S. M. Yiu, and B. Luo, “Defend against internet censorship in named data networking,” in *2016 18th International Conference on Advanced Communication Technology (ICACT)*, pp. 300–305, Jan 2016.

[76] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, "Naming in content-oriented architectures," in *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking*, ICN '11, (New York, NY, USA), pp. 1–6, ACM, 2011.

[77] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev, and L. Zhang, "An overview of security support in named data networking," *IEEE Communications Magazine*, vol. 56, pp. 62–68, November 2018.

[78] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Commun. ACM*, vol. 43, pp. 45–48, Dec. 2000.

[79] WOT services, "Web of trust," 2007.

[80] Y. Yu, A. Afanasyev, Z. Zhu, and L. Zhang, "An endorsement-based key management system for decentralized ndn chat application," *University of California, Los Angeles, Tech. Rep. NDN-0023*, 2014.

[81] M. Mosko, E. Uzun, and C. A. Wood, "Mobile sessions in content-centric networks," in *Proceedings of IFIP Networking Conference and Workshops*, pp. 1–9, June 2017.

[82] A. Biryukov, I. Pustogarov, and R.-P. Weinmann, "Trawling for tor hidden services: Detection, measurement, deanonymization," in *2013 IEEE Symposium on Security and Privacy*, pp. 80–94, May 2013.

[83] The Tor Project, "Tor protocol specification," 2003.

[84] The Tor Project, "Tor metrics portal," 2019.

[85] M. Wright, M. Adler, B. N. Levine, and C. Shields, "Defending anonymous communications against passive logging attacks," in *2003 Symposium on Security and Privacy, 2003.*, pp. 28–41, May 2003.

[86] T. Elahi, K. Bauer, M. AlSabah, R. Dingledine, and I. Goldberg, "Changing of the guards: A framework for understanding and improving entry guard selection in tor," in *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*, WPES '12, (New York, NY, USA), pp. 43–54, ACM, 2012.

[87] W. Greene and B. Lancaster, "Carrier-grade: Five nines, the myth and the reality," in *Pipeline Volume3, Issue11*, Pipeline, 2006.

[88] NTT East Corporation, “Coverage areas of telephone exchange buildings.” https://www.ntt-east.co.jp/info-st/info_dsl/area.html, 2016.

[89] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos, “Information-centric networking for the internet of things: challenges and opportunities,” *IEEE Network*, vol. 30, no. 2, pp. 92–100, 2016.

[90] B. Raghavan, T. Kohno, A. C. Snoeren, and D. Wetherall, “Enlisting isps to improve online privacy: Ip address mixing by default,” in *International Symposium on Privacy Enhancing Technologies Symposium*, pp. 143–163, Springer, 2009.

[91] T. Lee, C. Pappas, and A. Perrig, “Bootstrapping privacy services in today’s internet,” *SIGCOMM Comput. Commun. Rev.*, vol. 48, no. 5, pp. 21–30, 2019.

[92] C. Hazay and Y. Lindell, “A note on the relation between the definitions of security for semi-honest and malicious adversaries.,” *IACR Cryptology ePrint Archive*, vol. 2010, pp. 1–4, 2010.

[93] G. T. Duncan and D. Lambert, “Disclosure-limited data dissemination,” *Journal of the American Statistical Association*, vol. 81, no. 393, pp. 10–18, 1986.

[94] A. Evfimievski, J. Gehrke, and R. Srikant, “Limiting privacy breaches in privacy preserving data mining,” in *Proceedings of the Twenty-second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 211–222, 2003.

[95] F. L. Hitchcock, “The distribution of a product from several sources to numerous localities,” *Journal of mathematics and physics*, vol. 20, no. 1-4, pp. 224–230, 1941.

[96] M. Ashwin, G. Johannes, K. Daniel, and M. Venkitasubramaniam, “L-diversity: privacy beyond k-anonymity,” in *22nd International Conference on Data Engineering (ICDE’06)*, pp. 24–24, 2006.

[97] D. L. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” *Communications of the ACM*, vol. 24, pp. 84–90, 1981.

[98] Ú. Erlingsson, V. Pihur, and A. Korolova, “Rappor: Randomized aggregatable privacy-preserving ordinal response,” in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067, 2014.

[99] B. Gedik and L. Liu, “Protecting location privacy with personalized k-anonymity: Architecture and algorithms,” *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, 2008.

[100] Q. Cui, N. Wang, and M. Haenggi, “Vehicle distributions in large and small cities: Spatial models and applications,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10176–10189, 2018.

[101] A. Meyerson and R. Williams, “On the complexity of optimal k-anonymity,” in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 223–228, 2004.

[102] G. Sun, V. Chang, M. Ramachandran, Z. Sun, G. Li, H. Yu, and D. Liao, “Efficient location privacy algorithm for internet of things (iot) services and applications,” *Journal of Network and Computer Applications*, vol. 89, pp. 3–13, 2017. Emerging Services for Internet of Things (IoT).

[103] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent development and applications of SUMO - Simulation of Urban MObility,” *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, 2012.

[104] L. Codecá, R. Frank, S. Faye, and T. Engel, “Luxembourg SUMO traffic (lust) scenario: Traffic demand evaluation,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 2, pp. 52–63, 2017.

[105] K. Kita, Y. Kurihara, Y. Koizumi, and T. Hasegawa, “Location privacy protection with a semi-honest anonymizer in information centric networking,” in *Proceedings of the 5th ACM Conference on Information-Centric Networking*, pp. 95–105, 2018.

[106] G. M. Morton, “A computer oriented geodetic data base; and a new technique in file sequencing,” tech. rep., IBM, 1966.

[107] O. Pele and M. Werman, “A linear time histogram metric for improved sift matching,” in *Computer Vision–ECCV 2008*, pp. 495–508, Springer, 2008.

[108] O. Pele and M. Werman, “Fast and robust earth mover’s distances,” in *2009 IEEE 12th International Conference on Computer Vision*, pp. 460–467, IEEE, 2009.

[109] C. A. Melchor, J. Barrier, L. Fousse, and M.-O. Killijian, “Xpir: Private information retrieval for everyone,” *Proceedings on Privacy Enhancing Technologies*, vol. 2016, pp. 155–174, 2016.

[110] D. Naylor, M. K. Mukerjee, and P. Steenkiste, "Balancing accountability and privacy in the network," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, (New York, NY, USA), p. 75–86, Association for Computing Machinery, 2014.

[111] H.-C. Hsiao, T. H.-J. Kim, A. Perrig, A. Yamada, S. C. Nelson, M. Gruteser, and W. Meng, "Lap: Lightweight anonymity and privacy," in *2012 IEEE Symposium on Security and Privacy*, pp. 506–520, IEEE, 2012.

[112] C. Chen and A. Perrig, "Phi: Path-hidden lightweight anonymity protocol at network layer," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 1, pp. 100–117, 2017.