Defending Against DDoS and Unauthorized Access Attacks in Information Centric Networking

by

Eslam Gamal Ahmed AbdAllah

A thesis submitted to the

School of Computing

in conformity with the requirements for

the degree of Doctor of Philosophy

Queen's University

Kingston, Ontario, Canada

March 2017

Dedication

This thesis is dedicated to

the soul of my lovely Mother, ***Habiba****.*

Abstract

Information Centric Networking (ICN) is a new communication paradigm that focuses on content retrieval from the network regardless of the content storage location. ICN changes the security model from securing the path to securing the content, which is available to all the ICN nodes as ICN users could benefit from any available copy. Existing security solutions cannot be applied directly to ICN architectures because of unique ICN attributes.

In this research, we propose a security framework for ICN traffic management that includes the required functions of three components: availability, access control, and privacy. Our framework has the following attributes: be integrated within the architecture; deliver contents with high availability; transfer contents securely to legitimate users; and preserve the privacy of ICN users and contents. In this thesis, we focus on the availability and access control components.

To build the proposed framework, it is crucial to have a comprehensive understanding of ICN attacks and their classification. In our research, we identify unique attacks to ICN architectures and other generic relevant attacks that have impacts on ICN. The attacks can be classified into four main categories: naming, routing, caching, and other miscellaneous attacks. We study the impacts of ICN attacks on ICN attributes and security services.

An attacker can easily send a large number of malicious requests or publish invalid contents or routes to cause Distributed Denial of Service (DDoS) and cache pollution.

Hence, we propose a solution for Defending Against DDoS in ICN routing and caching (DADI). DADI limits malicious requests, selects top-ranked contents and publishers, marks malicious routes, and caches the most popular contents. We evaluate DADI using various attack scenarios and under different ratios of attackers to legitimate users.

To prevent unauthorized access attacks, we propose Elliptic Curve based Access Control (ECAC) solution. In this protocol, fewer public messages are needed for access control enforcement between ICN subscribers and ICN nodes than the existing access control protocols. We perform security and performance analysis for ECAC. We evaluate ECAC using various scenarios and under different request rates and number of attackers with respect to the number of legitimate users.

Acknowledgments

First and foremost, I thank ALLAH the most gracious and most merciful for blessing me with guidance, ability, and patience to complete this work.

I would like to thank my Father, Gamal and my Mother, Habiba for their unconditional love, care, and support during all my life. You are the main reason for getting me up to this stage. I dedicate to you all success I achieve in my life. There are not enough words to express my feelings for you.

I would like to express my special appreciation and sincere gratitude to my advisors Prof. Hossam S. Hassanein and Prof. Mohammad Zulkernine, you have been great mentors for me. I would like to thank you for encouraging my research and for your constant guidance, enthusiasm, and patience. Your advice and help have been priceless. I would also like to thank Dr. Ahmed E. Hassan and Dr. Ying Zou for serving as my committee members and for their valuable feedback.

My thanks are also due to ALL the members of the Queen's Telecommunications Research Lab (TRL) and Queen's Reliable Software Technology (QRST) Laboratory for their friendship, collaboration, valuable discussions, and fun times. Special thanks to our great assistant Mrs. Basia Palmer for all her help and support.

I would like express appreciation to my beloved wife Sara Mohamad who spent sleepless nights with me and was always my support. I would also like to thank my lovely kids

Ibrahim and Habiba who made my life full of happiness. I like to thank my brothers Ahmed and Mohamed and my sister Marwa for their continuous encouragement and kind wishes for me. A special thanks to ALL of my family members. I cannot find any words to express how grateful I am to ALL of you. I would also like to thank ALL my teachers and friends who encouraged me to strive towards my goal.

During my PhD, I had research internships at Irdeto, which were great opportunities. I would like to express my thanks to ALL the members there and specially for Yuan Gu and Clifford Liem.

I would like to thank the School of Computing at Queen's University, Canada for this great opportunity and for offering a wonderful educational environment. I would also like to thank Natural Science and Engineering Research Council (NSERC), Mitacs, and Irdeto for their financial support. I would also like to thank my home country Egypt for everything that helped me to reach this point.

Statement of Originality

I hereby certify that all of the work described within this thesis is the original work of the author. Any ideas and/or techniques from the work of others are fully acknowledged in accordance with the standard referencing practices.

Eslam Gamal Ahmed AbdAllah

March, 2017

List of Acronyms

ICN	Information Centric Networking
NGI	Next Generation Internet
MDHT	Multilevel Distributed Hash Tables
DONA	Data Oriented Network Architecture
NetInf	Network of Information
NDN	Named Data Networking
PURSUIT	Publish Subscribe Internet Technology
API	Application Programming Interface
QoS	Quality of Service
DDoS	Distributed Denial of Service
CDN	Content Distribution Network
P2P	Peer-to-Peer
DNS	Domain Name System

DNSSEC	Domain Name System Security Extensions
RH	Resolution Handler
AN	Access Node
POP	Point of Presence
AS	Autonomous System
ACP	Access Control Policy
SSTP	Scalable and Secure Transport Protocol
DADI	Defending Against DDoS in ICN
FIB	Forwarding Information Base
PIT	Pending Interest Table
CS	Content Store
RIds	Resource Identifiers
SIds	Scope Identifiers
RN	Rendezvous Network
RI	Rendezvous Interconnect
TN	Topology Node
BN	Branching Node
RSR	Request Satisfaction Ratio

SE	Stretched Exponential
PLR	Positive Likelihood Ratio
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography
ECAC	Elliptic Curve based Access Control
DH	Diffie-Hellman
ITS	Intelligent Transportation Systems
DACPI	Decentralized Access Control Protocol for ICN
RFID	Radio Frequency IDentification
PKI	Public Key Infrastructure
ACED	Access Control Enforcement Delegation
SAC	Session-based Access Control

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 ICN Overview

According to Cisco Visual Networking Index [96], by 2020, there will be almost 4.1 billion Internet users and 26.3 billion network devices and connections globally, the average fixed broadband connection speed will increase to 47.7 Mbps, and IP video will represent 82 percent of all traffic. This increasing demand for highly scalable and efficient distribution of contents requires new alternative solutions for the upcoming Next Generation Internet (NGI), as the existing Internet architecture is becoming inadequate [1]. Information Centric Networking (ICN) is one of NGI alternatives, which focuses on contents rather than endpoints [15, 65]. ICN relies on unique attributes such as location independent naming, in-network caching, name-based routing and built-in security [10, 67].

The Internet was designed to exchange information between well-identified hosts. Nowadays, the Internet has changed into the Internet of Things, Internet of Services, Internet of People, and Internet of Media. In Internet of Things, objects and appliances are connected with their own IP addresses. Users can access different functionalities stored on different hosts in Internet of Services. The Internet has become Internet of People especially after

the explosion of social media and new web technologies. Media are shared and managed across different networks. The main concern in these new Internets is no longer endpoints, but contents. This future Internet comes with new requirements to efficiently serve the increasing number of users and objects.

Information Centric Networking (ICN) is one of the alternatives for these new Internets. The number of objects in ICN is expected to be several orders of magnitude higher than the number of nodes in current Internet architectures. Contents that can be stored anywhere and in any format are the main concern of new ICN communication paradigm. In ICN, securing the content itself is much more important than securing the infrastructure or the endpoints.

1.2 Problem Definition

ICN architectures are vulnerable to different types of attacks that affect ICN security and performance. ICN attacks can be classified into four categories: naming, routing, caching, and miscellaneous attacks. ICN content requests are visible to the network, which adds a greater threat with respect to privacy. Attackers try to censor Internet usage. ICN architectures increase the control of attackers on information flow and make blocking information much easier for them. In routing related attacks, malicious publishers and subscribers can publish and subscribe for invalid contents or routes. ICN caching is vulnerable to different kinds of attacks that pollute or corrupt the caching system, in addition to the difference between cached and uncached contents that can be used to violate ICN privacy. Other attacks are related to unauthorized access, and changing or deleting contents during transmission.

Existing security solutions cannot be applied directly to ICN architectures because of the unique ICN attributes which can be summarized as follows:

- Non-ICN security solutions are added as an overlay, while in ICN, security solutions should be integrated within the architecture.
- Current non-ICN architectures depend on IP-based addressing solutions, while ICN does not include any host addresses.
- Non-ICN architectures cache contents at specific caching points, while ICN depends on in-network caching, which is one of the major ICN components.
- In non-ICN architectures, requests can receive many responses, while in ICN, each request has one response and there is no response without a request.
- Existing access control mechanisms fail to address the following attributes of a good access control mechanism: 1) reduce unnecessary communication overhead, 2) eliminate modifications to ICN architectures, 3) minimize the exchange of secret keys, 4) do not require large number of extra operations for access control, 5) prevent access control attacks, 6) preserve the privacy of ICN users, 7) ensure the integrity of the retrieved content, and 8) can be applied to different ICN naming schemes.

1.3 Motivations

Although security is a major concern in ICN, the research efforts so far are designed for a specific architecture or specific attacks. Security researchers have only scratched the surface of security issues in ICN, and the research is in its early days. ICN security can be improved by detecting and preventing ICN attacks in an efficient way [8, 13, 27, 29, 34, 35, 37, 39, 64, 78, 84]. To achieve this goal, we need to develop a built-in security solution specifically for ICN architectures. This proposed research solution fulfills the most affected security services: availability, access control, and privacy; and it also maintains

the ICN functionalities: naming, routing, and caching. The availability component allows valid publications to be available to legitimate users. The access control component allows contents and users to be mutually authenticated. The privacy component preserves the privacy of both ICN contents and users.

In ICN architectures, an attacker can publish invalid contents or announce malicious routes and send malicious requests for available and unavailable contents. These types of attacks can cause Distributed Denial of Service (DDoS) and cache pollution in ICN architectures [8, 27, 28, 34, 37, 39, 84]. To address these attacks, we need a solution that detects and prevents these DDoS attacks.

The in-network caching attribute is one of the major differences between ICN and non-ICN architectures. In the current Internet architectures, contents are stored at specific points, which simplifies the access control mechanisms. Network security administrator can deploy their security modules in these specific points. In ICN, subscribers can access contents from different locations. In-network caching attribute makes the access control security service in ICN much more complicated [6, 11, 35, 38, 45, 54, 62, 70, 72, 80, 82]. We need a solution to address these unauthorized access attacks.

In this thesis, we also aim to find answers to the following questions:

1. What are the security vulnerabilities in ICN architectures?

2. How do these security vulnerabilities affect ICN security services?

3. Can we build security solutions that address the vulnerabilities in ICN architectures?

1.4 Thesis Contributions

The contributions of this thesis work can be summarized as follows:

Provide a taxonomy of ICN attacks [1]. The thesis provides a comprehensive taxonomy of ICN attacks based on ICN components. We evaluate the existing ICN attacks against security services and ICN unique attributes and identify the severity levels for each one of the ICN attacks.

Develop a security framework that can be implemented in ICN architectures for traffic management [2]. The proposed framework addresses three main problems. The first is distributed malicious publications and subscriptions that may negatively affect large-scale ICNs. The framework allows valid publications to be available to legitimate users. The second pertains to unauthorized access, which is exacerbated in ICN since content can be cached and accessed from any ICN node. The third is related to violating private information about either contents or users. Our proposed framework includes countermeasures for these three problems. The framework will be integrated within ICN architectures to achieve availability, access control, and privacy for ICN architectures.

Propose a solution to address DDoS attacks that affect ICN availability [3, 4, 5]. It consists of several countermeasures. The proposed solution Defending Against DDoS in ICN routing and caching (DADI) contains many functions to select the best available content and publisher, identify malicious routes, limit incoming malicious requests, and store the most popular contents. DADI has two phases: a detection phase and a prevention phase. In the detection phase, we compare user behaviors and ICN contents and publishers against identified threshold values and specify the attack case. In the prevention phase, we apply our countermeasures to mitigate these attacks. These countermeasures enable ICN routers to take appropriate actions against diverse attack scenarios based on some parameters. The threshold values of these parameters are determined by studying the behavior of legitimate users.

Propose a solution for ICN unauthorized access attacks [6, 7]. The suggested solution Elliptic Curve based Access Control (ECAC) provides mutual authentication between ICN users and nodes hosting the requested contents, in order to secure restricted-access contents. The main objective of ECAC protocol is to allow only legitimate users to access legitimate contents.

1.5 Thesis Outline

The remainder of this thesis is organized as follows.

- Chapter 2 discusses the necessity of ICN in the future Internet and introduces basic ICN components, architectures, and unique characteristics.
- Chapter 3 presents ICN attack taxonomy and their classification, shows the relationships between ICN attacks and unique ICN attributes, identifies the characteristics of ICN attacks and security services. It also discusses the impact of each attack on the services and calculates the severity levels for the attacks based on ten evaluation metrics.
- Chapter 4 introduces our reference model, assumptions, ndnSIM environment, and performance metrics. This chapter presents the proposed overall framework that includes the availability, access control, and privacy components.
- Chapter 5 describes different scenarios of routing and caching related DDoS attacks pertaining to ICN, presents our solution and its unique ICN features, and performs experiments for the suggested solution on ndnSIM. Our experiments show that the proposed solution effectively mitigates routing and caching attacks in ICN.

- Chapter 6 presents a Decentralized Access Control Protocol for ICN (DACPI) and an enhanced version that achieves better performance results based on elliptic curve (ECAC). The chapter presents the specifications, internal steps, and security and performance analysis for ECAC protocol. It defines attributes of good access control schemes and compares ECAC with respect to these attributes. This chapter also evaluates ECAC request access time delay. According to the security and performance analysis and experiments, ECAC prevents unauthorized access to ICN contents with minimum overhead.

- Chapter 7 concludes our work, discusses the limitations, and presents several future research directions related to ICN privacy, mobility-related attacks, and ICN integration with non-ICN architectures.

Chapter 2

ICN Components and Architectures

2.1 Introduction

Today's host-centric Internet architecture was designed between 1960 and 1970 with the simple goal of connecting a few computing devices across geographically distributed users. Historically, the Internet has developed in an incremental manner. As new requirements arise, incremental patches are introduced to handle these requirements. In the beginning, a Domain Name System (DNS) was developed after the Internet to resolve names to IP addresses. However, DNS failed to support content movement and location awareness, and that is why Content Distribution Network (CDN) was introduced later. Similarly, Peer-to-Peer (P2P) systems such as BitTorrent and Gnutella were designed to provide rapid content distribution and multi-source content retrieval. These solutions were added as overlays and could not impact the underlying network topology to achieve optimal performance, and that is why there is a need for new Internet architectures [58, 68].

The current Internet architecture is becoming inadequate for the new requirements of the future Internets. The Internet usage recently changed the Internet model from the legacy

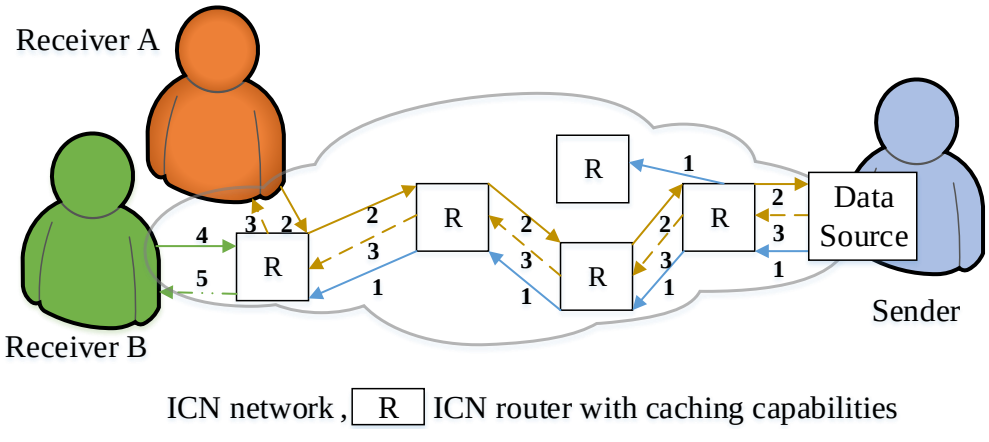

Figure 2.1: ICN basic process: 1- Publish message: A publisher sends a publication message with the content name to the ICN. 2- Subscribe message from subscriber A: A subscriber sends a subscription message with the content name to the ICN. 3- Delivery path for subscriber A: The ICN builds a delivery path from the data source to the subscriber. 4- Subscribe message from subscriber B: Another subscriber sends a subscription message for the same content. 5- Delivery path for subscriber B: The ICN delivers the content from the closest available copy via the ICN in-network caching

host-centric to content-centric architectures [20]. ICN is based on asynchronous publications and subscriptions. A publisher sends publication messages with content names to ICN to announce new contents to share. A subscriber sends subscription messages for contents using also content names. Publishers do not know who may be interested in their contents and subscribers do not know who have published these contents. ICN makes a delivery path between a publisher and a subscriber, when there is a match between a publication and a subscription. Then ICN content is transferred from a publisher to a subscriber. If there are other requests for the same content, ICN tries to deliver this content using in-network caching. Figure 2.1 shows the basic operation of an ICN architecture.

This chapter is organized as follows. Section 2.2 presents the main ICN components: naming, routing, caching, security, and application programming interface. Section 2.3 describes the most referenced ICN architectures: DONA, NetInf, NDN, and PURSUIT.

2.2 ICN Components

All ICN architectures have some generic concepts, which can be classified as follows: naming, routing, caching, security, and application programming interface. ICN uses the information object concept to refer to the content itself, which is the main focus of ICN, regardless of its storage location. ICN content may have different representations, and different copies for each representation.

2.2.1 Naming

The naming schemes in ICN can be classified into three categories: hierarchical, flat, and attribute-value pair based. In hierarchical naming, names are composed of multiple hierarchical components. A component can be any string of any length that is generated and assigned by users. The names in this category are human-friendly but non-persistent. In flat naming, which is also known as self-certifying naming, names consist of two parts of the form *P:L* and metadata. The first part *P* is the cryptographic hash of the owner's public key. The second part *L* is a content label assigned by the owner. Metadata contains the full public key and digital digest signed by the owner. Self-certifying names are unique, persistent, not limited to any organization and easy for integrity checking. In attribute-value pair based, each attribute has a name, a type and a set of possible values, but the names in this scheme do not ensure uniqueness or security for content names [29, 85].

2.2.2 Routing

In ICN, routing techniques can be classified into two major approaches: name resolution and name-based routing. Name resolution involves two steps. In the first step, the content name is resolved to a single or a set of IP addresses. In the second step, using any topology

based on shortest path routing like Open Shortest Path First (OSPF), the request is routed to one of these IP addresses. In the process of name-based routing, a request is routed directly based on the content name and state information is stored along the way, so that the content itself can be delivered using the reverse path to the subscriber [10].

2.2.3 Caching

In-network caching in ICN works based on the following principles: applied to all content delivered by any protocol (uniform), published by any content providers (democratic) and available to all network nodes (pervasive). ICN caching can be on-path and off-path. Each node caches any content passing through this node in on-path caching techniques. In off-path caching techniques, neighbouring nodes also can cache these contents [40, 93].

2.2.4 Security

In ICN architectures, as the network and/or user can use any available copy, security cannot be bound to the endpoints or storage location like host-centric architectures. Consequently, new information centric security concepts are required that let the security be applied to the content itself. ICN architectures integrate security aspects within the architecture itself not as an overlay on the routing layer.

2.2.5 Application Programming Interface (API)

An API in ICN is used to request and deliver the content. The source publishes its content to make it available for other users in the network. A user sends a subscription message for the content that he/she is interested in. The two operations (publish and subscribe) use the content name as the prime parameter.

2.3 ICN Architectures

The idea of ICN started in the TRIAD project in 2000 [24], after which a number of research projects appeared. In the following subsections, we will describe the most referenced ICN architectures.

2.3.1 Data Oriented Network Architecture (DONA)

The Data-Oriented Network Architecture (DONA) [53] project proposes a clean slate design of content naming resolution system. It proposes self-certifying naming scheme with a hierarchically organized name resolution infrastructure.

Naming: Names in DONA are of the form *P:L*. The *P* part of the name ensures provenance, and the signature in the metadata ensures content integrity. When a subscriber requests a content with the two parts, he/she receives the content, public key, and signature. The subscriber first compares the hash value of the received public key with the sent *P* to ensure content provenance. Then the subscriber compares the hash value of the decrypted received signature using publisher's public key with the signed hash value to ensure the content integrity.

Routing: DONA uses name resolution routing, in which the content name is resolved first to one or more addresses, then the request is forwarded to one of these addresses based on shortest path routing. DONA uses name resolution entities called Resolution Handlers (RHs). These RHs are organized in a hierarchal manner, and each administrative domain has one logical RH. DONA uses name resolution routing and offers two primitives FIND(P:L) and REGISTER(P:L). The "FIND" primitive is used to request contents, and the "REGISTER" primitive is used to publish contents.

2.3.2 Network of Information (NetInf)

NetInf proposes to use self-certifying names similar to DONA and a multi-level DHT-based name resolution service called MDHT. These MDHT are nested in Access Node (AN), Point of Presence (POP), and Autonomous System (AS) levels. Each DHT level can run its own algorithm. NetInf offers two primitives Get(X) and Put(X). The get primitive is used to request contents, and the put primitive is used to publish contents. The put primitive registers a content at the three levels, while NetInf tries to find the content level by level when a subscriber sends a get message [30, 101].

2.3.3 Named Data Networking (NDN)

NDN proposes a clean slate routing scheme by replacing IP address with content name and uses the content name as the universal transport component [46].

Naming: NDN uses a hierarchal naming scheme as the current URL naming. Names in NDN consists of many components, and each component can be a string of any length. The advantage of this scheme over the self-certifying scheme is that NDN names are human-friendly. However, from the Internet usage experience, people nowadays do not remember URL names and use search engines for requesting websites, which makes the user-friendly naming feature not so important when comparing it to the more secure self-certifying scheme.

Routing: NDN uses name-based routing to deliver contents to subscribers. In name-based routing, the request is routed directly based on content name and the content is returned back using the reverse path. Each NDN node contains the following tables: Forwarding Information Base (FIB), Pending Interest Table (PIT), and Content Store (CS). The FIB stores the possible data sources. The PIT stores the return path for contents and

the interfaces for the incoming requests. All incoming requests for the same content are stored in the same entry in the PIT table. NDN has two primitives: Interest and Data. The interest primitive is used by subscribers to request contents, and the data primitive is used by publishers to publish contents. Any NDN can satisfy the request, if it has a copy of the requested content. In NDN, no data is sent without an interest and each interest receives one data packet.

2.3.4 Publish Subscribe Internet Technology (PURSUIT)

PURSUIT proposes a different update of self-certifying naming scheme and name-based routing.

Naming: PURSUIT uses the self-certifying naming schemes and content names are called Resource Identifiers (RIds). PURSUIT also uses Scope Identifiers (SIds.), which identifies content access rights, authorization, reachability, availability, replication, persistence, and upstream resources. Content publication and subscription in PURSUIT are done using resource and scope identifies pairs (SId;Rid). There is an entity called data source that resides in the network edge responsible for content persistence. Data sources refresh content publication states in the network in a regular manner [55].

Routing: PURSUIT proposes a clean slate routing architecture consisting of four components: Rendezvous, Topology, Routing and Forwarding. The Rendezvous component consists of one Rendezvous Network (RN) per domain. The main job of RN is to locate the publications and scopes of its network. These RNs are connected on the Internet scale by Rendezvous Interconnect (RI), which performs like a middle layer between publishers and subscribers and matches requests with contents. Also, the topology component has one Topology Node (TN) per domain, which manages the intra-domain topology and load

Table 2.1: Host-centric vs. ICN

	Host-centric	ICN
Naming	define host topological position	define the content
Routing	between hosts using IP addresses	using name resolution entity or name based routing
Caching	specific caching points	each node can cache any content
Security	secure communication channels	secure the content itself
API	send data to a specific location	publish and subscribe contents

balancing. The routing component consists of many Branching Nodes (BN). BN uses the information acquired from TN to build up routing path and route subscriber requests to available data sources and cache contents. Also, the forwarding component consists of many Forwarding Nodes (FN), which implement efficient forwarding algorithms to send the requested content back to the subscriber.

2.4 What Makes ICN Unique?

ICN changes the current Internet architecture to fulfill new user expectations. Table 2.1 summarizes the important differences between host-centric and ICN architectures in terms of naming, routing, caching, security, and application programming interface. In addition to the preceding concepts, ICN as a solution for the upcoming Internet era should also achieve the following design principles [12, 98]:

- **Scalability**. Serve a very large number of entities.
- **Availability**. Ensure that the network has a usable operation rate.
- **Reliability**. Easily recover in case of any failures.

- **Network management simplicity**. Support self-configured and self-optimized networking.
- **Quality of Service (QoS)**. Develop prioritization criteria for contents that allow the network to provide QoS differentiation between these contents.
- **Loosely coupling system**. Provide more flexibility in time constraints, sequencing, and environment assumptions.
- **Flexible business models**. Allow and encourage different stakeholders to share and participate with their contents in the ICN open environment.

2.5 Summary

ICN is one of the candidates for the future Internet to satisfy the recent increasing demands of user requirements. ICN has unique characteristics that enable this new paradigm to compete and fulfill user expectations. In this chapter, we discuss self-certifying and hierarchical naming schemes, name-based and name resolution routing, in-network caching, built-in security, and publish/subscribe primitives. ICN has many proposals for architectures. In this chapter, we focus on the most popular four architectures: DONA, NetInf, NDN, and PURSUIT. The self-certifying naming scheme has been used in three of these four architectures and is considered to be a promising technique over the hierarchal naming.

Chapter 3

ICN Attacks: Taxonomy, Attributes, and Severity Levels

3.1 Introduction

In ICN, securing the content itself is much more important than securing the infrastructure or the endpoints. To achieve the security goals in this new paradigm, it is essential to understand the ICN attacks and their classification. In this chapter, we provide a survey of unique attacks to ICN architectures and other generic attacks that have an impact on ICN. It provides a taxonomy of these security attacks in ICN, which are classified into four main categories: naming, routing, caching, and miscellaneous attacks. Further, the chapter shows the relations between ICN attacks and unique ICN attributes and between ICN attacks and security services (availability, access control, and privacy). Finally, the chapter presents the severity levels of ICN attacks based on ten evaluation metrics. Specifically, this chapter addresses the following primary points:

A taxonomy of ICN attacks. To the best of our knowledge, this chapter proposes the first taxonomy of ICN attacks that classifies these attacks into four categories: naming, routing, caching, and miscellaneous attacks. Then it further classifies the attacks in each category based on the types of the attacks.

Relation between unique ICN attributes and ICN attacks. We study how the attacker benefits from unique ICN attributes: location independent naming, state decorrelation, in-network caching, and ubiquitous publication/subscription to perform his/her attack.

Relation between security services and ICN attacks. We address how each ICN attack affects the security services: availability, access control, and privacy.

Severity levels of ICN attacks. We calculate the severity level for each ICN attack based on the following evaluation metrics: block content retrieval, access user request, cache pollution, misrouting, request timeout, number of affected nodes, geographical distribution of attacked networks, remote exploitation, availability of attacked environment, and difficulty level of fixing damage. The calculation is based on the assumption that there is no explicit security mechanism used to defend against such attacks.

The survey presented in this chapter can aid to answer many important questions, which summarize our major contributions as follows:

- What are the most important attacks that may take place in an ICN environment?
- How do unique ICN attributes relate to ICN attacks?
- What are the most important security services in ICN?
- What are the most severe ICN attacks?
- Why do we need new security solutions in ICN architectures?

The remainder of the chapter is organized as follows. Section 3.2 presents a taxonomy of ICN attacks. Section 3.3 shows the relations between ICN attacks and unique ICN attributes. Section 3.4 discusses ICN attacks with respect to security services. Section 3.5 presents the severity level of ICN attacks. Finally, Section 3.6 draws conclusions.

3.2 Taxonomy of ICN Attacks

ICN has many security issues to be addressed. There are new types of attacks in ICN that did not occur before or did not have a significant impact in other environments. Additionally, many attacks that occur in other environments can also occur in ICN environments [32, 42, 44, 66, 69, 83]. This taxonomy classifies ICN attacks (new and legacy) into four categories as shown in Figure 3.1: naming, routing, caching, and miscellaneous attacks. This classification depends on the attacker's *main* target. Although each attack appears on only one category, it may impact other categories as well. For example, both flooding and unpopular request attacks affect ICN routing and caching. In a flooding attack, the attacker's main target is to overload and exhaust routing resources and as a consequence it affects the caching system. In unpopular request attacks, the attacker's main target is to violate cache relevance and as a consequence it affects the routing system. The proposed categories are briefly introduced in the following four paragraphs.

Naming related attacks. ICN architectures have a greater threat to the privacy as content requests are visible to the network. Many attackers try to censor/monitor Internet usage. An ICN architecture provides more access to user requests that would increase the attackers control on information flow and makes blocking information much easier for them. In the naming related attacks in ICN, the attacker tries to prevent the distribution of a specific content by blocking delivery of this content and/or by detecting who requests this content [13, 29].

Routing related attacks. ICN content delivery depends on asynchronous publication and subscription, which adds extra effort to ensure consistency among distributed data states. Some attacks like jamming and timing aim to fail this state consistency, which may lead to unwanted traffic flows and/or denial of service. Other attacks, like infrastructure and

flooding attacks, try to exhaust the resources like memory and processing power that are used to support, maintain and exchange content states. In addition, the infrastructure in ICN relies on the integrity and correctness of content routing, and is therefore threatened by poisonous injections of paths and names [8, 27, 34, 37, 78].

Caching related attacks. Caching is one of the important components in ICN as the performance of the ICN infrastructure is based on receiver driven caching that aims to deliver the closest available copy to the user. Therefore, ICN is vulnerable to all operations that pollute or corrupt the caching system [39, 64, 84].

Miscellaneous attacks. The threats in this category aim to degrade the ICN service and allow the attacker to make unauthorized access. These attacks lead to insufficient or erroneous data distribution [35].

In the following subsections, we describe the attacks, scenarios, and impacts of each one of the four categories. The discussed attacks can also be classified as follows: new attacks in ICN such as bogus announcements and time analysis attacks; legacy attacks with new scenarios and a greater impact in ICN such as naming, routing related attacks, uniform, random, and unpopular requests in caching related attacks; legacy attacks with a different impact in ICN as described as the other attacks.

3.2.1 Naming Related Attacks

The attacks in this category can be classified into watchlist and sniffing attacks. In ICN, the network nodes can access user requests. The attacker uses this attribute in addition to location independent naming to perform these types of attacks. There is a generic assumption that the attacker who compromises an ICN node or router can access it and monitor requesters [13].

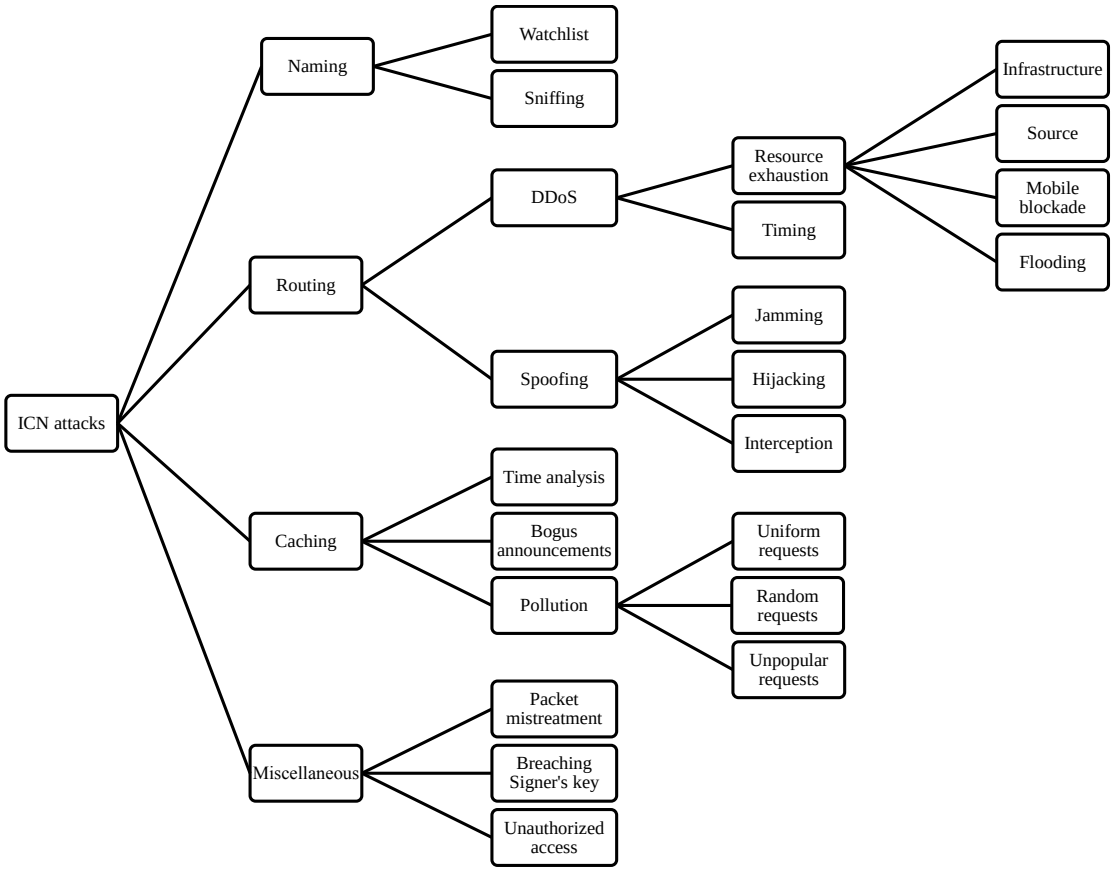

Figure 3.1: Taxonomy of ICN attacks

Watchlist. An attacker has a predefined list of content names that he/she wants to filter or delete. Then the attacker monitors network links to perform a real-time filtering. The attacker may delete the request and/or record the user who requested that data, in case of any matching against the predefined list. In addition, the attacker may try to delete the matched content itself. As depicted in Figure 3.2, the attacker captures user requests to filter and record who requested what. The attacker also filters and records return contents, which contain information about the publisher and the data. The filtration is based on the attacker's predefined list.

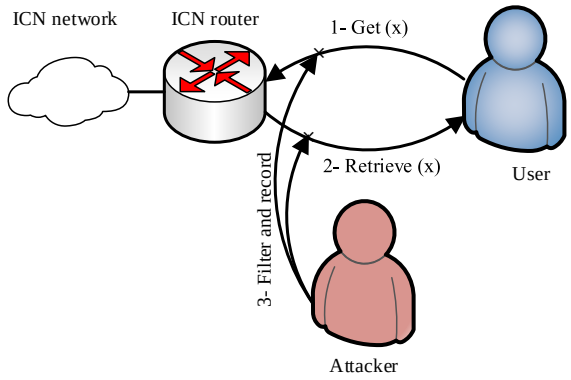

Figure 3.2: Watchlist attack: 1- A user requests for ICN content named (x). 2- Normally, the user should retrieve the content (x). 3- An attacker can filter and record the requests and/or the contents based on his/her predefined list

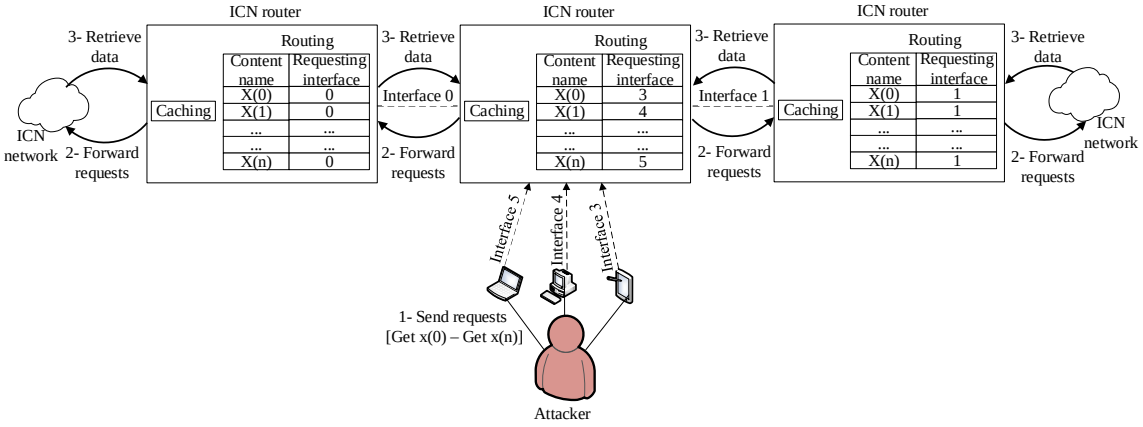

Figure 3.3: Infrastructure attack: 1- An attacker, who controls many end systems, sends a large number of requests to ICN routers. 2- The attacked routers forward these requests to the neighboring routers, and in turn they send it to their neighboring routers and so on. 3- ICN starts to retrieve these large amounts of data from different paths and sends it back to the requested locations

Sniffing. Unlike the predefined list in the watchlist attack, the attacker monitors the network to check the data if it should be marked in order to filter or eliminate it. The data should be marked if it contains the specified keywords. The attack scenario is the same as the watchlist attack. The remarkable difference is that the attacker does not have a predefined list, but he/she makes some analysis on requests or on contents.

The impact of naming related attacks

- **Censorship**. Using the naming related attacks, any attacker can censor the contents that he/she wishes.
- **Privacy**. Using these types of attacks, the attacker can monitor the content requests of a large number of users and record who made these requests. The ICN network accesses user requests, which results in a worse privacy situation.
- **Denial of service**. The attacker prevents user requests of the marked content, leading to unanswered requests.

3.2.2 Routing Related Attacks

The attacks in this category can be classified into distributed denial of service (DDoS) and spoofing attacks. DDoS can be classified into resource exhaustion and timing attacks. Resource exhaustion can be classified into infrastructure, source, mobile blockade, and flooding attacks. Spoofing attacks can be classified into jamming, hijacking, and interception attacks.

Infrastructure. An attacker sends a massive number of requests for available/unavailable contents. As ICN architectures try to find the closest copy from the best available location, these requests take different routes towards the source causing overload conditions. If the number of these requests is significantly high, it leads to a denial of service. This attack may be further amplified, as regular users after certain timeout send retransmission requests. Similar to the hijacking attack, this threat can be mitigated because routing mechanisms in ICN attempt to route towards multiple locations. As illustrated in Figure 3.3, the attacker, who controls many end systems, sends a large number of requests to one or more

ICN routers to fill the routing table and exhaust processing and memory resources. As a consequence, the attacked routers forward these requests to the neighboring nodes, which in turn forward it to the next neighboring nodes and so on. If the number of invalid requests is so high, any legitimate request takes a longer response time. Consequently, if the response time exceeds the request timeout, then the request may not be answered. This scenario can lead to denial of service or at least long delays.

Source. In ICN, attacking a single source may also lead to overload conditions for the routing infrastructure. An attacker sends a large number of requests to or from a specific content source to degrade its performance. As a consequence, this attack increases the response time of content delivery for this content source or its access router. In addition to this effect, the attack can lower the data return rate and affect all requests at all nodes on paths to receivers. The attack scenario is the same as the infrastructure attack. This attack affects not only the attacked source, but also the overall network.

Mobile blockade. A mobile attacker can overload a region by traversing neighboring networks on circular paths, while sending an enormous number of content requests. The attacker aims to overload the mobile access routers to make it exceed the state timeout, which leads to a blockade of the regionally available networks. The retransmission of requests is part of the mobility nature in an ICN environment, which adds difficulty in detecting this attack [77]. The attack scenario presented in Figure 3.4 is the same as the infrastructure attack. The difference is that the mobile attacker sends an enormous number of requests to neighboring networks, whereas he/she is traversing between the networks in a circular and continuous manner.

Flooding. The existing solutions for flooding attacks in ICN are designed to limit the number of request states which are not appropriate for ICN [37, 87, 88, 89]. An attacker

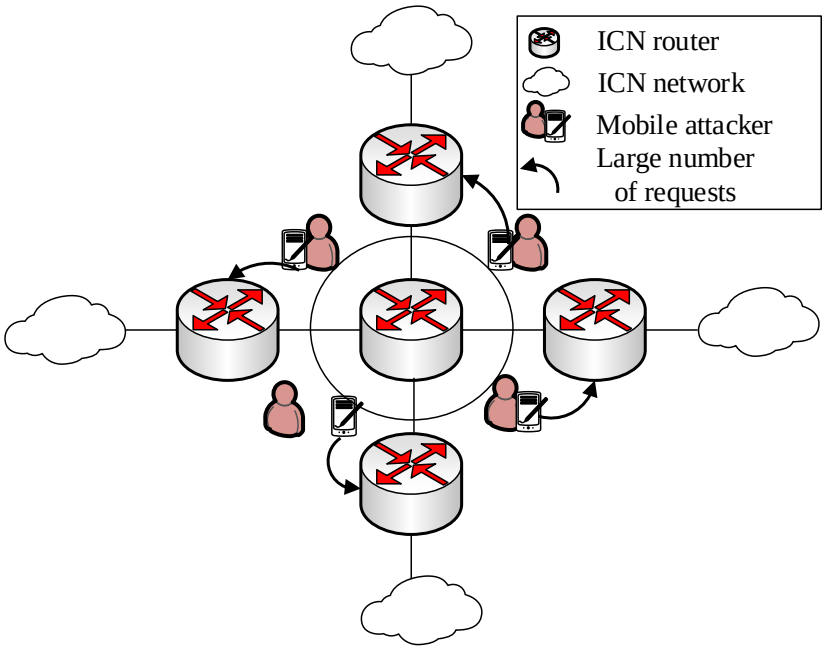

Figure 3.4: Mobile blockade attack: A mobile attacker sends a large number of requests, while he/she is traversing ICN neighboring networks

can easily create an interest bulk that exceeds this limit. The attacked node accepts a certain number of requests, and then ignores the remaining requests. As a consequence, the attacker succeeds to overload the overall infrastructure and harms all proximate users. The attack scenario is the same as the infrastructure attack. The difference is that the attacker sends a number of requests that exceeds the limits of the ICN nodes, and therefore ICN neglects the legitimate requests directed to the attacked nodes.

Timing. This refers to increasing the request timeout for some ICN nodes, to violate the consistency between the ICN asynchronous publication and the subscription process. An attacker sends a large number of requests to degrade the performance of some routers, so that request routing and data forwarding exhibits longer delays. The attack scenario is also the same as the infrastructure attack. The difference is that the attacker sends a large number of requests on one or more routes to increase the request timeout for legitimate

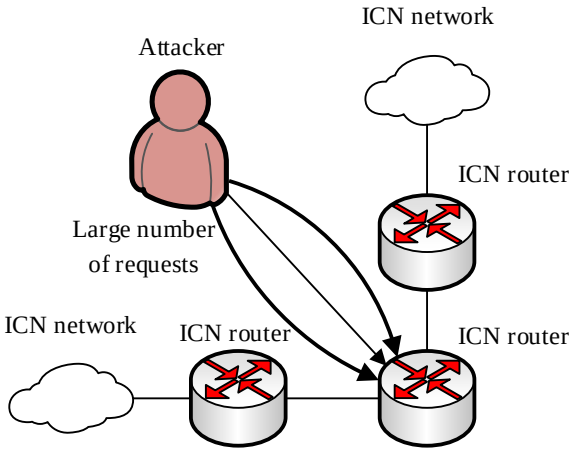

Figure 3.5: Jamming attack: An attacker sends a large number of requests to a shared node

user's requests.

Jamming. A node on a shared link sends a large number of malicious unneeded content requests. The attacker who masquerades as a trusted subscriber sends the malicious requests to disrupt the information flow in the system. The ICN network replies and content is sent to the destination without a receiver. This attack scenario is the same as the infrastructure attack. The difference, as presented in Figure 3.5, is that the attacker sends requests to a shared node, which forwards it to neighboring nodes.

Hijacking. Unlike host-centric architectures, any node in ICN can cache and publish or subscribe for contents. An attacker who masquerades as a trusted publisher may announce invalid routes for any content. Content requests from users in the proximity of the attacker are directed towards these invalid routes. Consequently, these requests will be unanswered, which lead to a DoS. The effect of this attack may be exacerbated, if the attacker has the ability to hijack invalid routes on a large scale. The effect of this attack is lessened because the routing mechanisms in ICN attempts to route towards multiple locations. As depicted

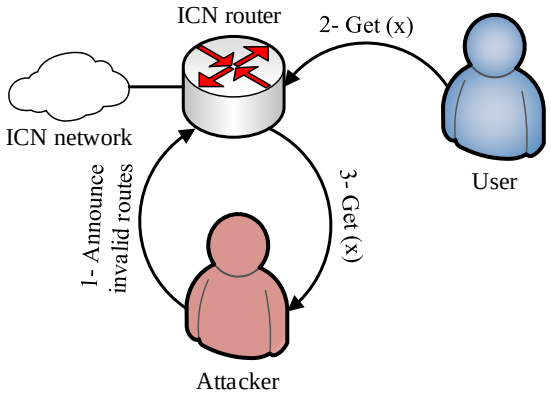

Figure 3.6: Hijacking attack: 1- An attacker announces invalid routes for some content including (x). 2- A user requests for ICN content named (x). 3- ICN router redirects the user's requests to the attacker's malicious routes, and consequently the user does not get any response

in Figure 3.6, the attacker announces invalid routes for some contents to attract the user requests. When legitimate users send requests for one of these malicious routes, ICN nodes forward these requests to the malicious node. Consequently, the legitimate user does not receive a response.

Interception. This attack is similar to the usual "man in the middle" attack. Unlike a hijacking attack, an attacker who masquerades as a trusted publisher announces invalid routes, while maintaining a record of valid routes to the content. Content requests can then be captured and sent to the proper location. Although the receiver gets the content normally, the attacker gains knowledge of the requested content. As shown in Figure 3.7, the attacker announces invalid routes for some contents to attract user requests. When legitimate users send requests for one of the malicious routes, ICN nodes forward these requests to the attacker's malicious node. The attacker records who requested this content and then forwards it to get the actual data. When the actual data arrives to the attacker's node, the attacker forwards it back to the requested ICN node, which in turn forwards it to

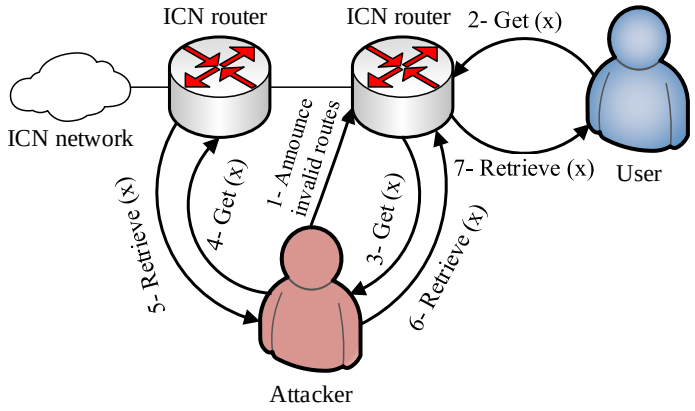

Figure 3.7: Interception attack: 1- An attacker announces invalid routes for some content including (x). 2- A user requests for ICN content named (x). 3- ICN router redirects the user request to the attacker's malicious routes. 4- The attacker forwards the request to get the actual content. 5- The attacker retrieves the content (x). 6- The attacker forwards the content to the requested user. 7- The user retrieves the content (x)

the legitimate user. For the user, the scenario seems to behave normally, but actually the attacker violates the user's privacy.

The impact of routing related attacks

- **Denial of service**. DoS can occur due to many attacks in this category, like in sending many requests for unavailable contents or to a single source, mobile blockade, flooding, hijacking, and timing. Consequently, intermediate timers delete requests with the expired timeouts, which may lead to DoS or at least long delays.

- **Resource exhaustion**. There are many sources for resource exhaustion in the ICN infrastructure that come from misuse or uncontrolled traffic such as sending a large number of requests and flooding attacks.

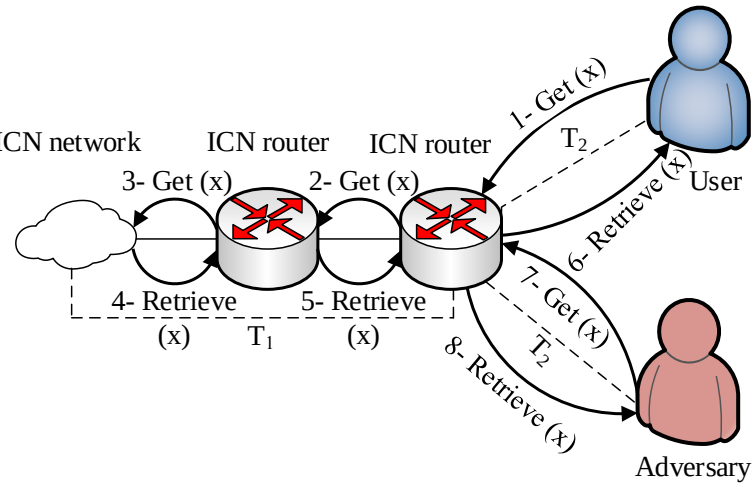

Figure 3.8: Time analysis attack: 1- A user requests for ICN content named (x). 2 and 3- ICN routers try to find the content (x). 4 and 5- ICN routers forward the content (x) to the requested user. 6- The user retrieves the content (x) in total time T_1+T_2. 7- An adversary requests for the content (x). 8- The adversary retrieves the content (x) in time T_2 only, as routers cache the content

- **Path infiltration**. In ICN, copies of content are typically distributed to many untrusted locations, and therefore it is difficult to authenticate valid origins for contents. Hijacking and interception are the major sources of path infiltration in ICN as attackers may announce invalid routes and claim them as trusted ones.

- **Privacy**. The privacy violation in the interception attack gives the attacker unauthorized access to user requests especially when the attacker is topologically close or on the route to the user.

3.2.3 Caching Related Attacks

The attacks in this category can be classified into time analysis, bogus announcements, and cache pollution attacks. The cache pollution can be classified into uniform, random, and unpopular request attacks.

Time analysis. In ICN, any node can potentially cache any content. An adversary measures the time difference between request response times for cached and uncached content. This difference can be used to conclude if a proximate user has previously requested the same content as the requested one by the adversary. This attack violates the user's privacy as the adversary can gain information about this proximate user. As depicted in Figure 3.8, T_1 is the time required to send the request and receive data between the content source and the closest router to the user or the adversary, and T_2 is the time required to send the request and receive data between the user or the adversary and the closest router. When a legitimate user requests a certain content, the ICN infrastructure forwards the request to the content source and returns the data to the user in a total time (T_1+T_2). Then if the adversary requests the same content, he/she gets it in time T_2 as there is already a cached copy of content. The adversary uses this time difference to know if a proximate user requested this content before or not.

Bogus announcements. As the caching system is a major part of the ICN architecture, an attacker can send many announcement updates for content or cached copy at a frequency that exceeds the local content request routing convergence time, to violate the caching and routing systems. As a consequence, an ICN will not be able to match the legitimate requests with these network quick updates. These overloaded announcements lead to incomplete and erroneous content retrieval as illustrated in Figure 3.9.

Uniform requests. An attacker aims to spoil the ICN in-network caching system and to change the content popularity. The attacker forces ICN caches to store unpopular contents by sending uniform requests for these unpopular contents. Unpopular content refers to a content that is not frequently requested. Alternatively, the attacker may request false contents to fill the caches with invalid contents. A content is fake if it is modified, or does

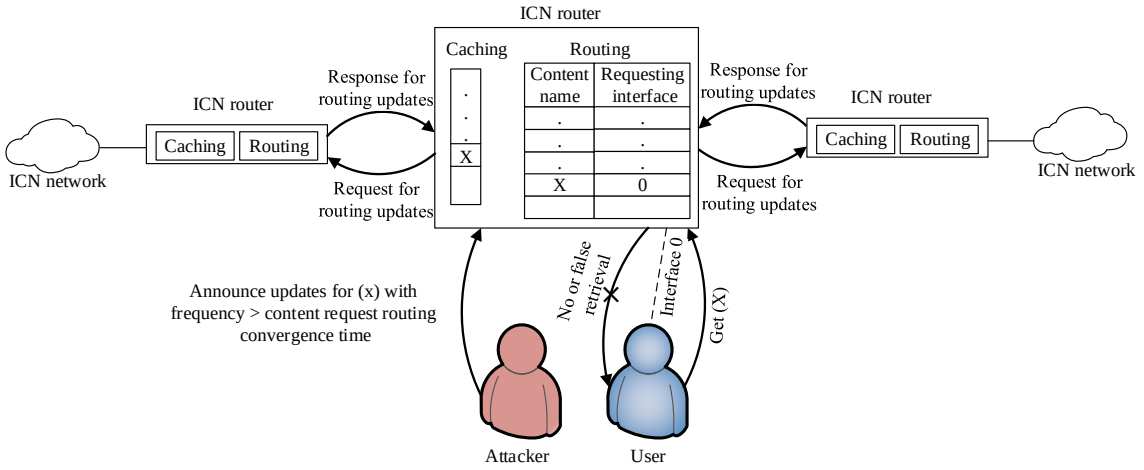

Figure 3.9: Bogus announcements attack: A user requests for ICN content named (x), while an attacker sends a large number of updates for contents including (x), with a frequency that exceeds content request routing convergence time. ICN routers will not be able to update its routing table because of these bogus announcements, which lead to no or false content retrieval

not come from the intended source, or it is not the content requested by the user. As shown in Figure 3.10, in the normal case if a second user requests a cached copy, he/she gets the data from the closest available location as the caching system caches each content passing through it. As shown in Figure 3.11, in the attacked case, the attacker sends a massive number of uniform distribution requests to spoil the caching system. In the latter case, if the second user requests the same content, his/her request takes the full path as the first user.

Random requests. An attacker sends a large number of random requests to change cache popularity. For an attacker, sending random distribution requests is easier than the uniform or unpopular cases.

Unpopular requests. An attacker only requests unpopular contents to spoil the ICN in-network caching and change the content popularity. This attack requires a prior knowledge of the content popularity.

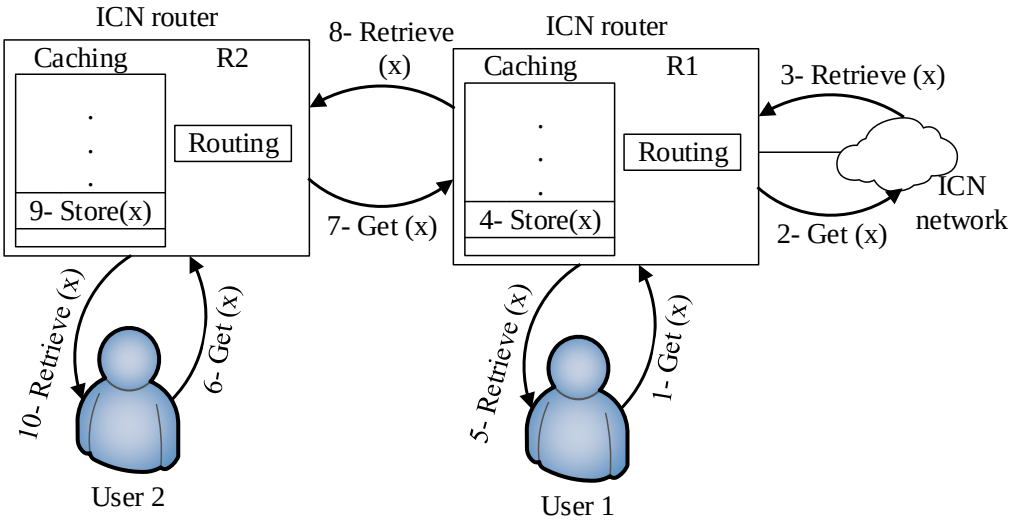

Figure 3.10: Uniform requests attack (normal case): 1- User 1 requests for ICN content named (x). 2- R1 router tries to find the content (x). 3- R1 retrieves the content from ICN network. 4- R1 caches the content (x). 5- User 1 retrieves the content (x). 6- User 2 requests the same content (x) via R2 router. 7- R2 tries to find the closest copy, which exists in R1 router. 8- R1 sends the content to R2 router. 9- R2 caches the content (x). 10- User 2 retrieves the content (x)

The impact of caching related attacks

- **Privacy**. The caching mechanism in ICN is uniform, democratic and pervasive, which causes a greater privacy risk than in current architectures. As in the time analysis attack, the adversary can know whether a proximate user has previously requested this content or not, which violates user's privacy.

- **Denial of service**. Bogus announcements cause many updates to contents, which lead to incomplete or erroneous data states. The mapping system will not be able to process these updates and, as a consequence, users do not retrieve the required contents.

- **Cache pollution**. Any user in ICN can send many uniform, random, and unpopular

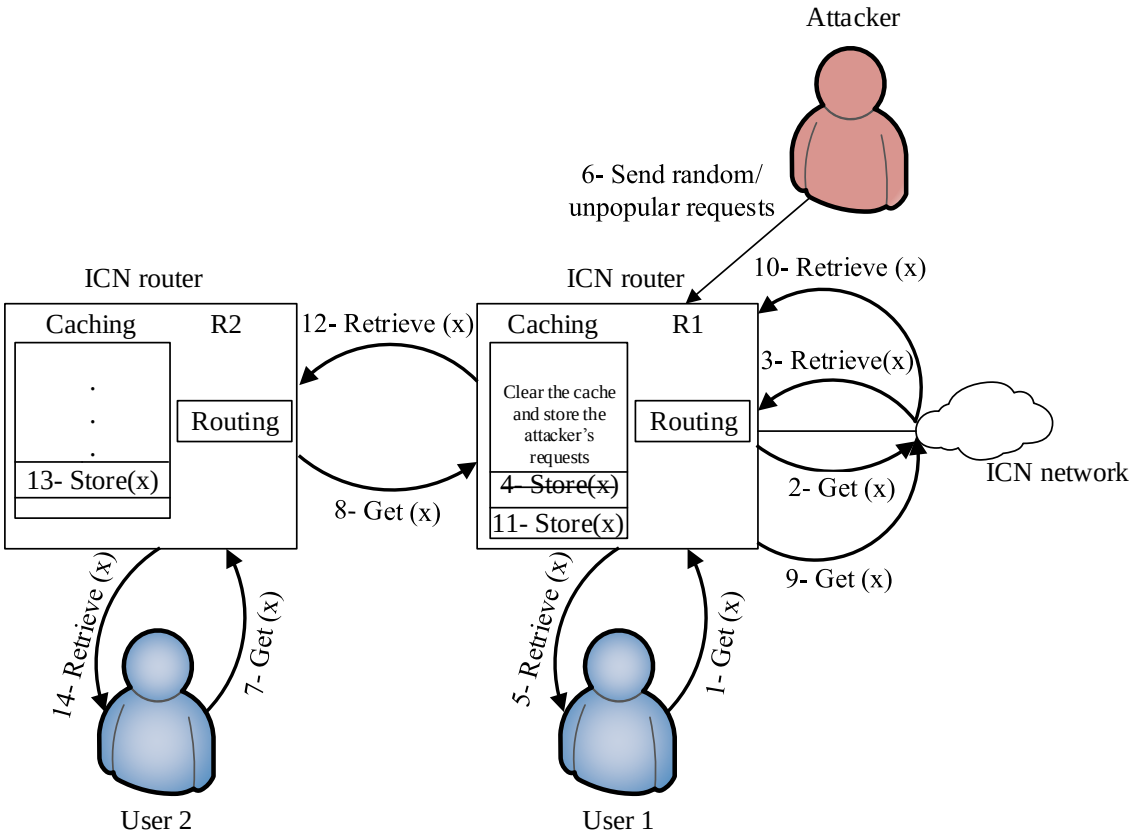

Figure 3.11: Uniform requests attack (attacked case): 1- User 1 requests for ICN content named (x). 2- R1 router tries to find the content (x). 3- R1 retrieves the content from ICN network. 4- R1 caches the content (x). 5- User 1 retrieves the content (x). 6- An attacker sends a large number of uniform/random/unpopular requests to violate the cache. 7- User 2 requests the same content (x) via R2 router. 8- R2 tries to find the closest copy and sends request to R1. 9- R1 router tries to find the content (x). 10- R1 retrieves the content from ICN network. 11- R1 caches the content (x). 12- R1 sends the content to R2. 13- R2 caches the content (x). 14- User 2 retrieves the content (x)

requests, which cause cache pollution [36].

3.2.4 Miscellaneous Attacks

The other attacks can be classified into packet mistreatment, breaching signer's key and unauthorized access attacks.

Packet mistreatment. This refers to normal active network attacks during data transmission that also includes the replay attacks. An attacker, who has access to a link fraudulently or maliciously, tries to block, change, or reply to requested data many times. For the attack scenario, the attacker access ICN nodes or network links to do the following: modify packets during transmission or, reply it many times to requester or, generate content on behalf of a legitimate user.

Breaching signer's key. An attacker can use any common attack to breach the signer's keys. This problem with ICN has a greater impact as publishers sign contents that are available for a long time and in large volumes. As shown in Figure 3.12, the attacker retrieves some contents from a specific publisher to break the publisher's key. The data contains publisher public information and signature. This data may be large enough to simplify the attacker's task to get the signer's key.

Unauthorized access. An attacker can access a certain content sent to a specific user or group of users that he/she is not allowed to access. In ICN, unauthorized access attacks become easier because an attacker can use any available copy for a content, which is distributed in different network locations.

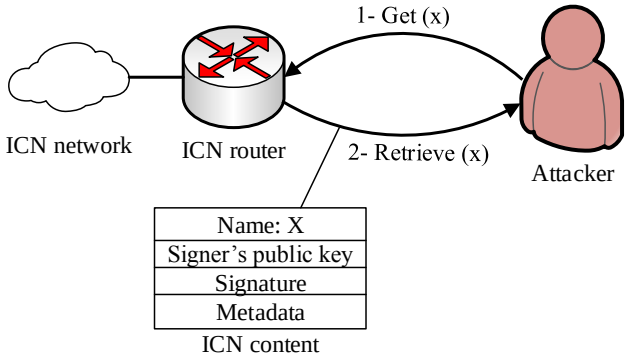

Figure 3.12: Breaching signer's key attack: 1- An attacker requests for ICN content named (x). 2- The attacker retrieves the content (x) that contains signer's public key and signature, which can be used with the content itself to determine the signer's key

The impact of miscellaneous attacks

- **Congestion**. The attacker redirects the packets to heavily loaded links, which can lead to congestion in the network. In addition, packet mistreatment attacks can result in lowering of the connection throughput.

- **Denial of service**. The attacker sends a large number of packets toward a source or network entity causing DoS attacks using packet mistreatment attacks.

- **Masquerading**. The attacker claims that he/she is a trusted entity. If the attacker succeeds to get the signer's key, then he/she can intercept, analyze, and/or corrupt the communications.

- **Unauthorized access to data**. In ICN, routers have direct access to content requests; therefore if the attacker succeeds to hack a router, then he/she is able to monitor the requests submitted by the users. This allows the attacker to discover user requests and monitor the user's daily life. For example, the attacker might track a certain user

by capturing his/her requests.

3.3 ICN Attributes Affecting Security

We identify four attributes, which may assist in increasing the impact of attacks in the ICN. By using these attributes, the attackers may be able to focus on attacks that have more consequence or are harder to detect or prevent in ICN. These attributes are as follows:

- **Location independent naming**. This attribute allows content retrieval from multiple unknown or untrusted locations. ICN requires a secure naming system to name contents regardless of its location.

- **State decorrelation**. ICN has two asynchronous states: request routing and content delivery. ICN requires consistency between these two states. Failures in the state consistency may lead to a DoS or unwanted traffic.

- **In-network caching**. Caching is one of the prominent characteristics of ICN architectures. Any node of the network can cache any item that passes through it. The content can be delivered from the closest cache that contains the content instead of going to the hosting server.

- **Ubiquitous publication/subscription**. Any user can access ICN network from any location and act as content suppliers or content consumers. Users may send unwanted contents or requests.

Table 3.1 shows the relation between ICN attacks and these ICN attributes. In this table, we show how the attacker can benefit from these attributes to achieve each attack. There are two values: P for primary attribute; and S for secondary attribute. These values indicate

how much the attacker uses each attribute in each attack (P: an attacker depends completely on this attribute, and the attack cannot happen without this attribute; S: an attacker depends partially on this attribute and uses it as an aid for an attack).

Table 3.1: ICN attributes affecting security (P: primary; S: secondary; Blank: No impact)

Attack	Location independent naming	State decor-relation	In-network caching	Ubiquitous publication/ subscription
Watchlist	P			
Sniffing	P			
Infrastructure		P	S	P
Source		S	S	P
Mobile blockade		S	S	P
Flooding		S	S	P
Timing		S	S	P
Jamming		S	S	P
Hijacking	P		S	P
Interception	P		S	P
Bogus announcements	S	P	P	P
Uniform requests		S	P	P
Random requests		S	P	P
Unpopular requests		S	P	P
Time analysis			P	
Packet mistreatment		S	S	S
Breaching signer's key	S	P	S	S
Unauthorized access	S		P	S

Table 3.2: ICN attacks vs. security services (H: high; M: medium; L: low; Blank: No impact)

Attack	Availability	Access control		Privacy
		Confidentiality	Integrity	
Watchlist	L	L		H
Sniffing	L	L		H
Infrastructure	H			
Source	L			
Mobile blockade	L			
Flooding	M			
Timing	M			
Jamming	L			
Hijacking	L	L		
Interception		L		M
Bogus announcements	H			
Uniform requests	M			
Random requests	M			
Unpopular requests	L			
Time analysis				M
Packet mistreatment	M		H	
Breaching signer's key		H		H
Unauthorized access		M	M	

3.4 ICN Attacks vs. Security Services

Because of the nature of the ICN architectures, ICN has greater availability, unauthorized access, and privacy risks than current networking paradigms, and there is an urgent need for a new security solution that detects/prevents all these attacks. The solution must achieve the three security services: availability, access control, and privacy. Availability ensures

that objects published in the network have to be available and accessible for authorized entities. We consider confidentiality and integrity as access control security services. Confidentiality indicates only eligible entities can access secured information. Data integrity means the ability to identify any accidental or intentional changes to information objects and the corresponding metadata. Privacy represents the protection of users as well as content privacy. Table 3.2 shows the relation between ICN attacks and security services. In this table, we show the effect of each attack on the security services using the OWASP risk rating [99]. The following values are used: H for high, M for medium, and L for low.

- **Availability** (service loss). H: all services completely affected; M: extensive primary services affected; L: minimal services affected.
- **Confidentiality** (data disclosure and sensitivity). H: all data affected; M: extensive critical data affected; L: minimal non-sensitive data affected.
- **Integrity** (data corruption). H: all data totally affected; M: extensive serious data affected; L: minimal data affected.
- **Privacy** (reveal of personal identifiable information). H: any user; M: proximate users; L: one individual.

3.5 Severity Levels of ICN Attacks

According to severity assessment by Symantec [100], we define ten metrics to evaluate the severity level of each attack. Some of these metrics related to the ICN architecture such as block content retrieval, access user request, cache pollution, and request timeout. The other metrics generally evaluate the effect of each attack on the attacked environment. In

assessing the severity of these attacks, we assume that there is no explicit security mechanism for these attacks for the ICN considered in this chapter. The severity level for each attack can be determined by the following metrics:

- **Block content retrieval**. H: blocks contents directed to any ICN user; M: blocks contents to neighboring network users; L: blocks contents to proximate users.
- **Access user request**. H: accesses requests from any ICN user; M: accesses requests from proximate users; L: accesses requests from specific users.
- **Cache pollution**. H: affects all cache entries; M: affects large number of cache entries; L: affects limited and small number of cache entries.
- **Misrouting**.
- **Request timeout**.
- **Number of affected nodes**.
- **Geographical distribution of attacked networks**.
- **Remote exploitation**.

The preceding five metrics are assessed as follows: H: attack targets/affects/controls large scale ICN networks; M: attack targets/affects/controls neighboring networks; L: attack targets/affects/controls specific nodes.

- **Availability of attacked environment**. H: attacker has no constraints; M: attacker attacks certain locations; L: attacker should have some privilege or prior knowledge.

- **Difficulty level of fixing damage**. It depends on five attributes: misrouting, number of affected nodes, geographical distribution of attacked networks, remote exploitation, and availability of attacked environment. H: cannot recover from the impact if the attacker accesses private information or at least three of the five dependent attributes are high; M: at least three of the five dependent attributes are medium; L: anything else.

Table 3.3 shows the severity level of ICN attacks, which can be classified into high, medium, and low severity. We calculate the severity for each attack by assigning a numeric value for each level (0 for no impact; 1 for low; 2 for medium; 3 for high). Then we add the values for each attack and calculate the percentage of the attack severity. The final severity level is low, if the attack has an effect of less than or equal to 30%; medium, if the attack has an effect of more than 30% and less than or equal to 70%; and high, if the attack has an effect of more than 70%. The final severity reflects the impact of each attack on the ICN environment. Figure 3.13 shows the severity of each attack. The high severity level of attacks such as the infrastructure and bogus announcements means that these attacks cause a catastrophic effect on the ICN environment because they can be performed in a distributed manner on a large scale network to affect any user. The low severity level of attacks such as hijacking, time analysis, and breaching signer's key means that these attacks cause a minor effect on the ICN environment because of their limited influence on the networks and users. In between, the medium severity attacks (e.g., watchlist, flooding, and random requests) cause a partial negative effect on the ICN environment.

Table 3.3: ICN attacks severity (H: high; M: medium; L: low; Blank: No impact)

Attack	Block content retrieval	Access user request	Cache pollution	Misrouting	Request timeout	Number of affected nodes	Geographical distribution of attacked networks	Remote exploitation	Availability of attacked environment	Difficulty level of fixing damage
Watchlist	H	H			L	L		L	H	H
Sniffing	H	H			L	L		L	H	H
Infrastructure	H		H	H	H	H	M	H	H	H
Source	L		L	L	L	L		M	L	L
Mobile blockade	L		L	L	L	L		M	L	L
Flooding	M		M	M	L	L	L	M	M	M
Timing	M		M	M	L	L	L	M	M	M
Jamming	L		L	L	L	L		M	L	L
Hijacking	L		L		L	L		L	L	L
Interception		M	L		L	L		L	L	H
Bogus announcements	H		H	H	H	H	H	H	H	H
Uniform requests			M	M	L	M	H	M	M	M
Random requests			M	M	L	M	H	M	M	M
Unpopular requests			L	L	L	L	H	L	L	L
Time analysis		M			L	L		L	L	H
Packet mistreatment	L		M	M	M	M	L	M	M	M
Breaching signer's key					L			H		H
Unauthori-zed access		M			M	L		H	H	M

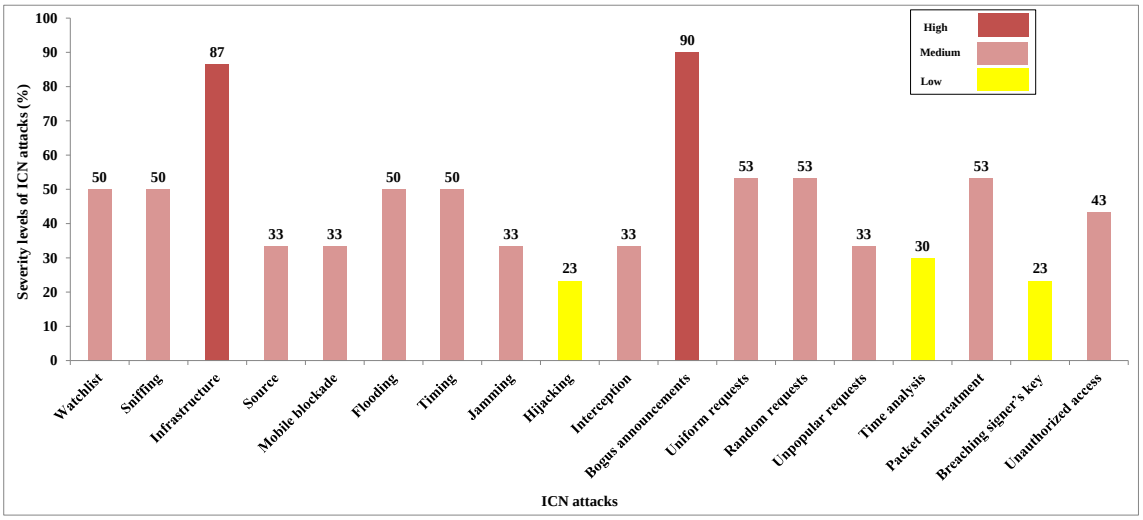

Figure 3.13: Severity levels of ICN attacks

3.6 Summary

In this chapter, we present four major aspects that can summarize our contributions. First, we develop a taxonomy of ICN attacks and classify the attacks into four categories: naming, routing, caching, and miscellaneous attacks. We describe each attack and the impacts of each category. Second, we derive the relationships between ICN attacks and unique ICN attributes. We show for each attack how the attacker depends on the corresponding attributes to perform his/her attack. Third, we derive the relationships between ICN attacks and security services and discuss the impact of each attack on the services. Fourth, we calculate the severity levels for the attacks based on ten evaluation metrics.

The attacks in an ICN environment can also be viewed from the following perspectives:

- New attacks in ICN environments. These include bogus announcements and time analysis attacks.

- Attacks that occur in non-ICN environments, but manifest themselves differently in

ICN with new scenarios and a greater impact. These are naming, routing related attacks, uniform, random, and unpopular requests in caching related attacks.

- Attacks that occur in both non-ICN and ICN environments in the same way but with a different impact. These are mentioned as "miscellaneous attacks" in this chapter.

Based on the analysis of the relationships between attacks, ICN attributes and security services, availability, access control, and privacy are the most affected services in ICN architectures:

- **Availability**. Sending massive malicious requests to the routing or caching systems in ICN are the main sources that affect the availability in ICN.

- **Access control**. Contents are cached and can be accessed from different locations. This simplifies unauthorized access risks.

- **Privacy**. Accessing user requests and the time difference between the cached and uncached content are the main impacting factors for the privacy in ICN.

Existing solutions target a specific architecture or specific types of attacks. Developing a complete security solution that can be applied in ICN architectures and integrated with the other technologies has become an urgent task for the ICN security. The major challenges for ICN security can be summarized as follows:

- Detection and prevention mechanisms for the attacks in this chapter should be an integral component of the architecture. Security in ICN must be attached to the content itself, as the content may be distributed in different locations. Any user can access any available copy, which causes unauthorized access risks.

- There are higher privacy risks in ICN than the other environments. We presented different types of attacks that can violate the privacy in ICN.

- Malicious publication/subscription is a risk as ICN is an open environment. We identified several ways invalid requests can be sent to overload the ICN network and exhaust resources.

Chapter 4

System Models and Proposed Framework

4.1 Introduction

Given the vulnerabilities that ICN have with regards to different types of attacks there is a need for new solutions for ICN architectures. This chapter presents the ICN reference model that we use in our work and the proposed framework. Our proposed solutions are based on some selected assumptions listed in this chapter. We also describe the simulation environment with brief coding guidelines and our performance metrics. Finally, the overall framework that consists of three main components is presented. The main functions of each of the components are also described.

4.2 System Models

4.2.1 ICN Reference Model

Our reference model consists of ICN routers, distributed storage location, and ICN users. ICN routers have routing and caching capabilities. The distributed storage locations are used to store the rating for ICN contents and publishers. ICN users are classified into publishers and subscribers. ICN subscribers can send a subscription message or vote against

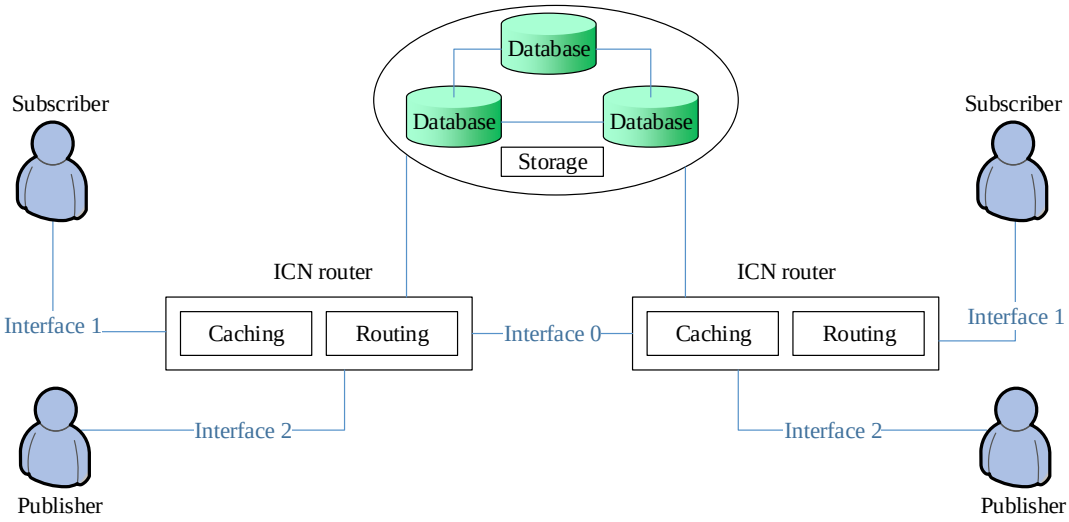

Figure 4.1: Generic ICN reference model

an invalid content. An attacker can be a malicious subscriber or publisher or both. The impact of these attacks can be amplified if the attackers act in a distributed manner. Attackers who control many end systems can cause DDoS attacks on a large scale ICN. ICN provides in-network caching with three main attributes. First, the caching is applied to all contents delivered by any protocol. Second, the caching is available to all network nodes. Third, the caching is available to all publishers. Figure 4.1 depicts a generic model for the ICN architecture that we use as a reference model.

The attacks and countermeasures in this thesis work are designed for the self-certifying naming scheme. For self-certifying naming scheme, an attacker needs an extra step to generate content names and sends malicious requests automatically, by searching for content names first. In the hierarchical naming scheme, an attacker can directly generate a large number of malicious requests, because the naming scheme has a certain structure.

4.2.2 Assumptions

In designing our framework and proposed solutions, we make the following assumptions:

- Depending on ICN self-certifying naming scheme: Our proposed protocols are based on ICN self-certifying naming scheme, which is a promising technique in ICN. DONA, NetInf, and PURSUIT architectures are using this naming scheme. Access control protocol based on self-certifying naming does not need to check ICN content integrity and publisher authenticity because they are verified in this naming scheme.

- Backbone network is secure: For access control protocols, we assume that our protocols will be applied in ICN edge routers because these edge routers are accessible by users. The aim of this assumption is to minimize extra authentication messages. In this case, required authentication messages are needed between ICN users and edge routers.

4.2.3 Simulation Environment

We evaluate the solution using ndnSIM, which is a simulator for the NDN architecture within NS-3 [9]. We build our experiments using backbone AT&T network, which is an Internet-like architecture. In our experiments, the network consists of 150 subscribers, 10 publishers and more than 40 routers as shown in Figure 4.2. In our work, we use many network scenarios to test our proposed solutions. We change subscriber request rates, subscribers request available and unavailable, popular and unpopular contents with different percentages. ICN cache sizes are also changed. The remaining part of this subsection shows the important functions and coding in ndnSIM that we have used.

To create nodes:

```
NodeContainer nodes;
nodes.Create (no_of_subs + no_of_routers);
```

To connect nodes using two links:

```
PointToPointHelper p2p;
```

To connect an subscriber to a router:

```
p2p.Install (nodes.Get (0), nodes.Get (0 + no_of_subs));
```

To connect two routers:

```
p2p.Install (nodes.Get (0 + no_of_subs), nodes.Get (1 +
no_of_subs));
```

To install ndn stack on all nodes:

```
ndn::StackHelper ndnHelper;
ndnHelper.SetDefaultRoutes (true);
// Detect PIT implementation and size
ndnHelper.SetPit("ns3::ndn::pit::Persistent", "MaxSize",
"1000");
// Detect cache replacement policy and size
ndnHelper.SetContentStore ("ns3::ndn::cs::Lru", "MaxSize",
"500");
ndnHelper.InstallAll ();
```

To install subscriber node:

```
ndn::AppHelper consumerHelper ("ns3::ndn::ConsumerCbr");
// Subscriber will request /prefix/0, /prefix/1, ...
consumerHelper.SetPrefix (consumerprefix);
```

```
consumerHelper.SetAttribute ("Frequency", StringValue
("10")); // 10 interests a second
consumerHelper.Install (nodes.Get (0));
```

To install publisher node:

```
ndn::AppHelper producerHelper ("ns3::ndn::Producer");
// Publisher will reply to all requests starting with
/prefix
producerHelper.SetPrefix (producerprefix);
producerHelper.SetAttribute ("PayloadSize", StringValue
("1024"));
producerHelper.Install (nodes.Get (1));
```

Tracing the rate in bytes and in number of packets of Interest/Data packets forwarded by an NDN node:

```
L3RateTracer::InstallAll ("drop-trace.txt", Seconds (0));
```

To obtain statistics of cache hits/cache misses on simulation nodes:

```
ndn::CsTracer::InstallAll("cs-trace.txt", Seconds(1));
```

To obtain data about for delays between issuing Interest and receiving corresponding Data packet:

```
ndn::AppDelayTracer::InstallAll("app-delays-trace.txt");
```

To show the simulation network, we use the netanim program:

```
AnimationInterface anim ("ndnSIM_ATT.xml");
```

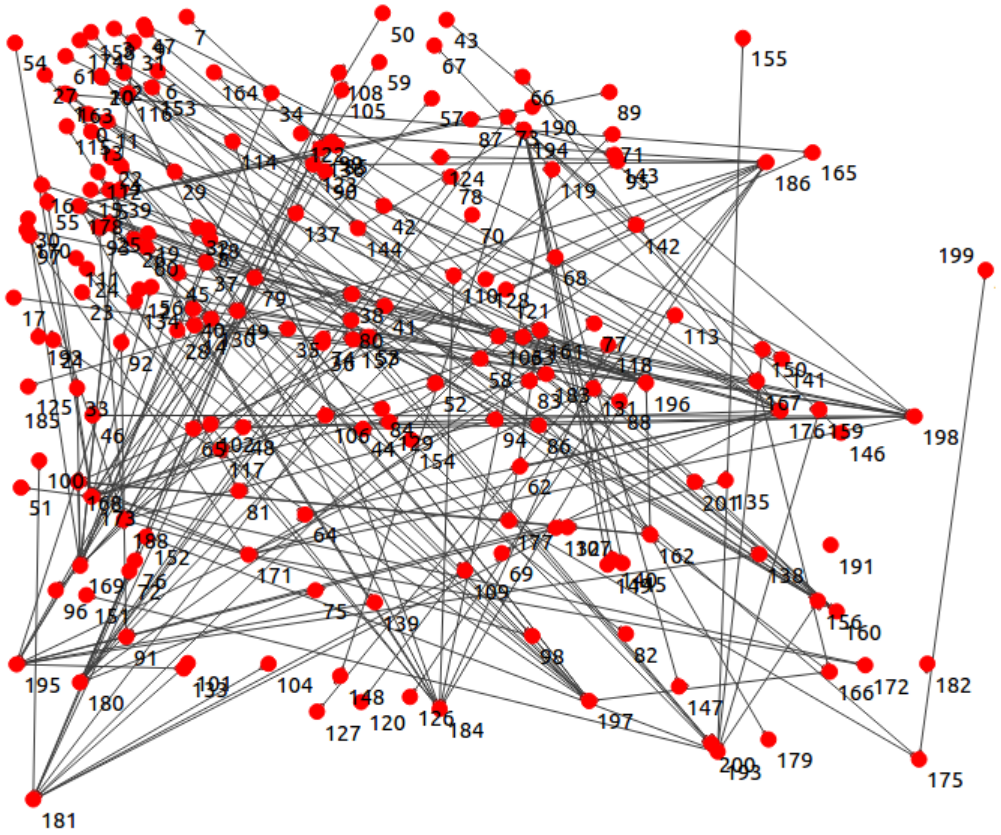

Figure 4.2: AT&T network: 150 subscribers, 10 publishers, and more than 40 routers

4.2.4 Performance Metrics

We compare our results with existing pioneer solutions and baseline scenarios. In the results, the following terminologies are used:

- Baseline: measurement for a certain metric in an ideal case scenario, when there are no attacks and without any add-on solution.

- Positive likelihood ratio (PLR): ratio between true positives and (1 - false positives). True positives indicate the ratio between attackers identified truly as attackers and the total number of attackers. False positives indicate the ratio between legitimate users identified as attackers and the total number of legitimate users. A 100% baseline

for PLR means that all attackers are correctly identified without any legitimate users identified as attackers.

Performance Metrics

Satisfied legitimate requests. The number of satisfied requests with respect to outgoing requests for legitimate users. This metric is chosen to study the impact of routing attacks with and without the proposed solution on legitimate users.

Cache hit ratio. The number of cache hits with respect to the number of outgoing requests. This metric is chosen to study the impact of caching attacks and the capability of our solution.

Request access time delay. It represents the delay between the first interest sent and the data packet received. This metric is chosen to study the impact of unauthorized access attacks and the capability of our solution.

4.3 Proposed Framework

4.3.1 Overview

The main objective of this framework is to study the ICN traffic to differentiate between legitimate and malicious user behaviors and hence provide appropriate countermeasures. Figure 4.3 shows the four main attack categories: naming, routing, caching and miscellaneous attacks and the relations between these attacks and ICN components and security services [1].

There is no comprehensive security framework that addresses the existing main ICN challenges, which can be summarized in the following points:

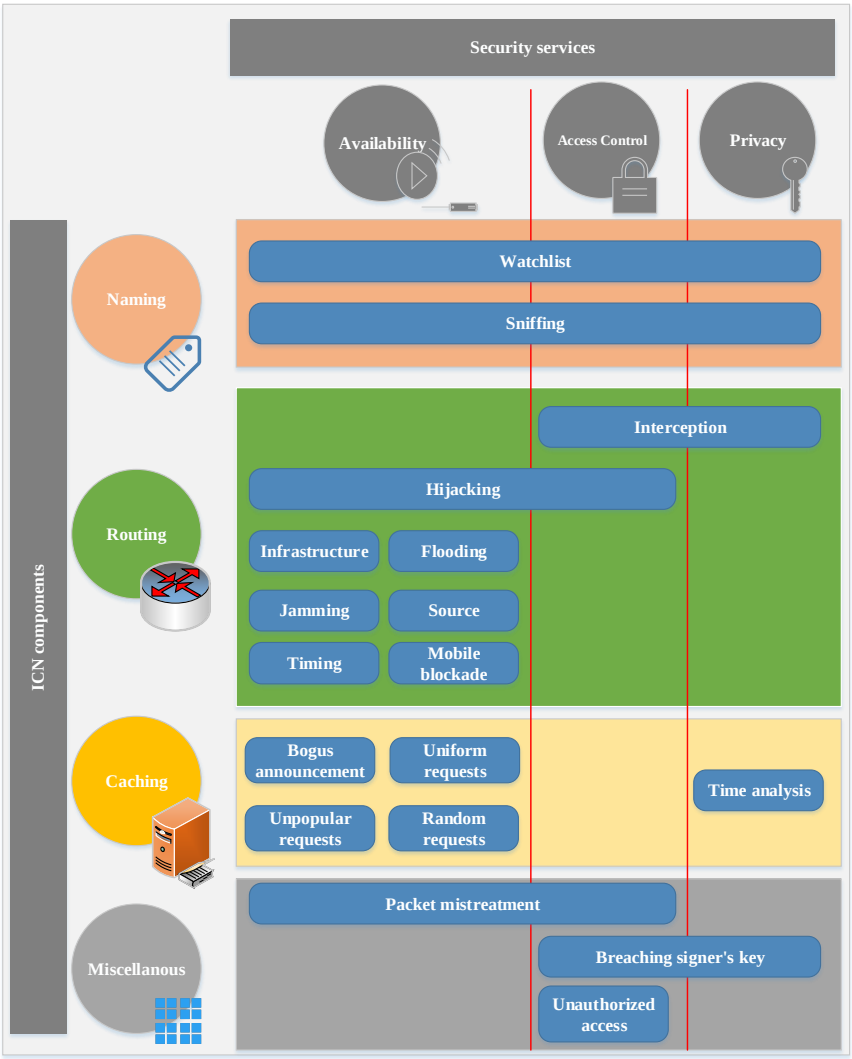

Figure 4.3: ICN attacks vs. security services: ICN naming, routing, caching, and miscellaneous attacks and their impact on availability, access control, and privacy security services

c1. Attackers send a large number of malicious requests for available and unavailable contents.

c2. Attackers announce malicious routes and send invalid contents.

c3. Attackers send various request patterns to force ICN caches to store unpopular contents instead of the popular ones.

c4. Attackers use in-network caching to get restricted-access contents, which are available in many distributed locations.

c5. Attackers change or send other contents during transmission.

c6. Attackers monitor content names and subscriber requests to filter, block and record private information about these contents and requests.

c7. Attackers measure the difference between cached and uncached contents to violate subscribers' privacy and get content popularities.

c8. Attackers break signer's cryptographic keys to behave as legitimate publishers.

This section presents the proposed security framework that includes the three components (availability, access control, and privacy) as depicted in Figure 4.4. The purpose of the availability component is to deliver ICN contents with high availability. Access control component goal is to deliver contents securely to legitimate users only. Privacy component aims to preserve the privacy of ICN users and contents. This framework will also be integrated with the major ICN components: naming, routing, and caching. There are some common functions that should be applied in all cases in this framework [1]. In the following

[1]The functions in red color are not implemented in this thesis and are left as a future work (See Section 7.2).

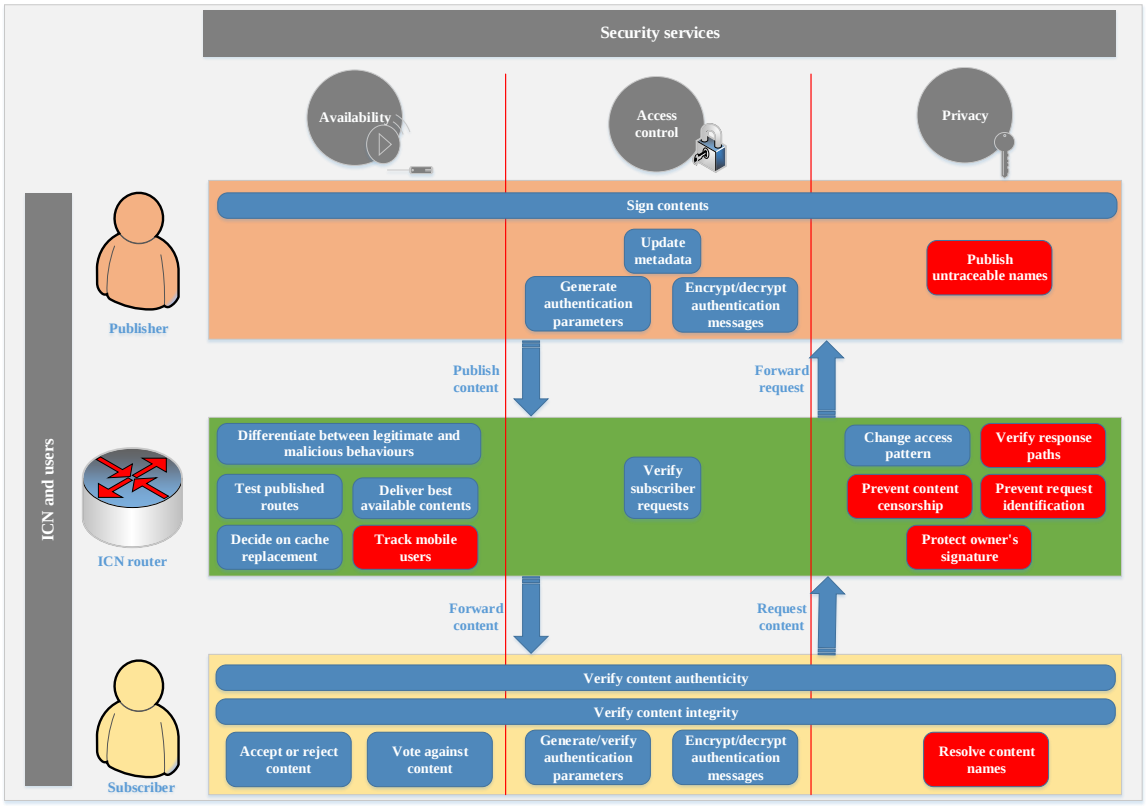

Figure 4.4: ICN security framework: required security functions at publisher, router, and subscriber ICN entities to achieve availability, access control, and privacy services

subsections, we write the associated challenge for each function between parentheses. For example, **c1** refers to the first aforementioned challenge.

Sign contents. A publisher signs ICN content so that a subscriber can verify the content authenticity and integrity. (c2, c5)

Verify content authenticity. A subscriber compares between the sent and received publisher information to ensure that the content is coming from the intended publisher. (c2, c5)

Verify content integrity. A subscriber compares between received signed hash value and

subscriber calculated hash value to ensure that no one modified the content during transmission. (c2, c5)

In the following subsections, we present the three components in detail.

4.3.2 Availability Component

An attacker's main goals are to cause DDoS to ICN architectures and cache pollution. An attacker can be a malicious publisher or subscriber. This component detects and prevents ICN DDoS related attacks to maintain the network availability for legitimate users. Also, it preserves the most popular contents in ICN caching in order to deliver contents efficiently [3, 4]. There exist previous works for ICN security that target DDoS attacks. Afanasyev et al. [8] propose three mitigation strategies to handle request flooding DDoS attack for unavailable contents and recommend satisfaction-based pushback mechanism as the best mitigation technique. In [84], Cacheshield uses a shield function that determines whether to cache contents or not at ICN routers with respect to random requests for ICN caching. Based on these related work examples and others [1, 27, 28, 34, 37, 39, 63, 82], the main functions that should be included in this component can be summarized as follows:

Differentiate between legitimate and malicious behaviors. This function evaluates incoming traffic requests and detects legitimate and malicious ones. Detection parameters include subscriber behavior, request rate, and pattern. (c1, c3)

Test published routes. ICN routers send a test message for the published route and based on received response, ICN routers mark this route as malicious or offer another opportunity to resend subscriber request. (c2)

Deliver best available contents. This function sorts ICN contents and publishers to enable an ICN router to select the best available content and publisher. This function depends on

parameters that should include subscriber request and feedback, publisher behavior, and content popularity. (c2)

Decide on cache replacement. ICN caching replaces least popular content with most popular ones, if the cache is full. ICN caching popularity parameters should include subscriber requests and number of requests for each content. (c3)

Track mobile users. This function tracks ICN mobile users in different ICN networks and detects their behaviors so that ICN routers can apply the above functions on these mobile users. (c1, c3)

Accept or reject content. Each subscriber has the right to accept or reject contents based on the received content and publisher. If the subscriber rejects the content, the subscriber can vote against it. (c2, c5)

Vote against content. Subscribers send voting messages against invalid contents to decrease their ranking. As the number of voting messages increase, the voting weight decreases for this user. (c2, c5)

4.3.3 Access Control Component

Nowadays, in the current Internet architecture, contents are cached at specific servers. This enables network security administrators to deploy their security modules and hence simplifies the access control mechanisms. ICN is an open environment, which enables subscribers to access contents from different locations because of the in-network caching attribute. This attribute makes the access control security service in ICN much more complicated than before. Access control component covers ICN confidentiality and integrity.

Fotiou et al. [35] propose a centralized access control mechanism (ACED) that evaluates subscriber requests against access control policies. In this mechanism, there are extra

entities such as access control provider and relaying party. Access control provider allows publishers to create contents with access control policy and evaluate subscriber requests. Relaying party distributes contents to legitimate users. Another centralized authentication and authorization mechanism for NetInf ICN architecture is presented in [11]. This mechanism uses ID-based cryptographic technique for securing the messages. This solution uses an extra entity named trusted ticket granting (TTG). TTG is responsible for key generation and distribution. Wang et al. [80] propose a decentralized session-based authentication mechanism that can be applied to different ICN architectures. In this mechanism, each ICN content has two names, one is public and the other is a secure name that is known only by legitimate users. AbdAllah et al. [6] propose a Decentralized Access Control Protocol for ICN (DACPI) that depends on ICN self-certifying naming scheme. DACPI uses RSA public key infrastructure, key exchange using Diffie-Hellman, hashing, and random number generations. We recommend decentralized mechanisms because they do not use extra entities, do not have a single point of failure, and can use fewer authentication messages. According to the aforementioned related work and others [1, 38, 54, 62, 70, 72, 82], the major functions that should exist in this decentralized access control component can be summarized as follows:

Update metadata. ICN publisher sends a publication message with the self-certifying content name. The metadata attached with the content is updated with the following information: hashing value of content and random numbers, secret information and cryptographic system public parameter. Cryptographic system encryption and decryption achieve message confidentiality, while shared secret key, hashing technique, self-certifying naming achieve content authenticity and integrity. (c4, c5)

Generate or verify authentication parameters. This function calculates and verifies the

required authentication parameters at publisher and subscriber sides. (c4, c5)

Encrypt and decrypt authentication messages. Messages are encrypted using receiver's public key and then using sender's private key to achieve authenticity and confidentiality. (c4, c5)

Verify subscriber requests. ICN routers verify subscriber requests by comparing subscriber hashing value with attached content hash value. (c4)

4.3.4 Privacy Component

The privacy component handles naming, interception, and time analysis attacks. An attacker's main goals are to censor contents and know their popularities. This component prevents attackers from knowing content popularities and private information about ICN users. In [63], the authors present malicious access privacy issues based on timing analysis attacks and related countermeasures. The time difference can be used as an indicator, if the subscriber has requested this content before or not. Wood et al. [82] personalize ICN cached contents to each ICN legitimate user by proposing encryption-based technique. Based on the existing work [13, 21, 45, 64], the essential functions that should be included in this component can be summarized as follows:

Publish untraceable names. Publishers use this function to publish names that cannot be tracked or expected by attackers. (c6)

Change access pattern. The purpose of this function is to prevent an attacker from differentiating between cached and uncached contents. ICN routers can respond with random delays or generate cache misses. (c7)

Verify response paths. An attacker needs extra time to censor and redirect the request

through certain paths. In this function, ICN routers calculate response time and if it exceeds a certain threshold, ICN routers neglect these routes. (c2, c6)

Prevent content censorship. This function generates dynamic mapping between actual content name and the submitted one from the subscriber. (c6)

Prevent request identification. This function removes any identification, if that exists, so an attacker cannot track who requested what. (c6, c7)

Protect owner's signature. This function protects owner's signature from cryptanalysts. (c8)

Resolve content names. This function enables a subscriber to generate the correct name for the intended content. (c6)

Table 4.1 presents and summarizes the relations between ICN attacks, framework components and functions [2].

4.4 Summary

In this chapter, we present our reference model, assumptions, and ndnSIM environment. We use satisfied legitimate requests, cache hit ratio, and request access time delay as performance metrics.

ICN comes with new challenges and requires new solutions rather than the existing ones. We propose a security framework for these attacks that contains availability, access control, and privacy components. The framework can be integrated within ICN architectures.

[2]The functions in red color are not implemented in this thesis and are left as a future work (See Section 7.2).

Table 4.1: ICN attacks, security services, and framework functions. The letters in parentheses indicate the function location (P: Publisher; R: ICN router; S: Subscriber)

ICN Attacks		Framework Function	Security Service
Naming	Watchlist	• Publish untraceable names (P) • Prevent content censorship (R) • Prevent request identification (R) • Resolve content names (S)	Privacy
	Sniffing		
Routing	Interception	• Verify response paths (R)	Privacy
	Hijacking	• Differentiate between legitimate and malicious behaviours (R) • Test published routes (R) • Deliver best available contents (R) • Accept or reject content (S) • Vote against content (S)	Availability
	Infrastructure		
	Source		
	Flooding		
	Jamming		
	Timing		
	Mobile blockade	• Track mobile users (R)	
Caching	Bogus announcements	• Same as routing solutions	
	Time analysis	• Change access pattern (R)	Privacy
	Uniform requests	• Decide on cache replacement (R)	Availability
	Random requests		
	Unpopular requests		
Miscel-laneous	Packet mistreatment	• Same as access control solutions	Access Control
	Breaching signer's key	• Protect owner's signature (R)	Privacy
	Unauthorized access	• Sign contents (P) • Update metadata (P) • Generate authentication parameters (P) • Encrypt/decrypt authentication messages (P,S) • Verify subscriber requests (R) • Verify content authenticity (S) • Verify content integrity (S) • Generate/verify authentication parameters (S)	Access Control

Chapter 5

Mitigating DDoS Attacks in ICN Routing and Caching

5.1 Introduction

ICN is an open environment that depends on in-network caching and focuses on contents. These attributes make ICN architectures subject to different types of routing and caching attacks. In ICN architectures, any user can publish or subscribe for contents. An attacker can publish invalid contents or announce malicious routes. He or she can send a large number of malicious requests for available or unavailable contents to overload an ICN architecture and fill up ICN interest tables. ICN routers forward these malicious requests to their neighbouring routers, which in turn send the malicious requests to their neighbouring routers. An attacker can also vote against valid contents to abuse the ICN architecture. These types of attacks can be carried out in a large distributed scale to cause Distributed Denial of Service (DDoS) for legitimate users [78].

ICN also depends on in-network caching, i.e., any node can cache contents to deliver contents to subscribers from the closest available copy. An attacker can send uniform, random, and unpopular requests to force ICN caches to store unpopular contents and evict the popular ones and hence an attacker pollutes ICN caching system.

The main contributions of this chapter are as follows:

- We provide a classification of DDoS attacks in ICN routing and caching.
- We propose a solution that covers malicious publications and subscriptions in addition to cache pollution attacks.

This chapter addresses the following attacks: invalid contents or malicious routes; fake requests for unavailable contents; large number of requests for available contents; requests for changing content popularities. An attacker can easily send a large number of malicious requests using any ICN architecture naming and routing schemes. Hence, our **D**efending **A**gainst **D**DoS in **I**CN routing and caching **(DADI)** is based on ICN generic components that can be applied to ICN architectures. DADI limits malicious requests, selects top-ranked contents and publishers, marks malicious routes, and caches the most popular contents. DADI has two phases: a detection phase and a prevention phase. In the detection phase, we compare user behaviors and ICN contents and publishers against identified threshold values and specify the attack case. In the prevention phase, we apply our countermeasures to mitigate these attacks. These countermeasures enable ICN routers to take appropriate actions against diverse attack scenarios based on some parameters. The threshold values of these parameters are determined by studying the behavior of legitimate users.

We evaluate DADI using various attack scenarios and under different ratios of attackers to legitimate users. We measure routing and caching performance metrics in the presence of DDoS attacks with and without DADI. Our results show that the solution mitigates these attacks and enhances ICN performance in the existence of these attacks. DADI maintains ICN availability for legitimate users and stores most popular contents in ICN caching in the presence of DDoS attacks.

The remainder of the chapter is organized as follows. Section 5.2 presents the related work in ICN routing and caching DDoS related attacks. Section 5.3 presents ICN routing and caching DDoS related attacks with their classifications. Section 5.4 presents our countermeasure parameters, and detection and prevention techniques for mitigating these ICN attacks. Section 5.5 shows the simulation results of the proposed solution. Finally, Section 5.6 draws some conclusions.

5.2 Related Work

There exist previous work that address some types of routing and caching attacks in ICN. Gasti et al. [37] present a high level classification of DDoS attacks and their solutions in Named Data Networking (NDN) architecture. For routing attacks, Afanasyev et al. [8] present request flooding DDoS attacks for unavailable contents. They propose three mitigation strategies based on Request Satisfaction Ratio (RSR) and recommend satisfaction-based pushback mechanism as a mitigation technique. Compagno et al. [27] also address request flooding DDoS attacks for unavailable contents. They define threshold values for RSR and Pending Interest Table (PIT) space in NDN architecture and then limit incoming requests based on these threshold values. In both solutions, an attacker can send requests for available contents to overcome these solutions or at least decrease the solution's effectiveness. Our solution for the routing part considers malicious publications and requests for available and unavailable contents. The solution is based on threshold values for RSR, request rate, cache hit ratio, rating for contents, and rating for publishers. These threshold values enable ICN routers to efficiently differentiate between legitimate and attack behaviors. Our rate limiting is based on attacker request rate, in addition to the threshold values.

For caching attacks, Conti et al. [28] present a lightweight mechanism for the detection

of cache pollution attacks in NDN. The detection is based on a threshold value calculated by the number of references for each content and their variations. The Cacheshield solution [84] handles random requests for ICN caching. Cacheshield uses a shield function that determines whether to cache contents or not at ICN routers. Cacheshield is tested on small scale networks and it stores content names and statistics about contents in ICN caching. Our solution is different from these solutions as follows: caching contents is based on decisions coming from our detection phase according to user behaviors and threshold values; ranking cached contents is based on their popularities and evicting the least popular ones in case of cache replacements.

Fotiou et al. [34] present a ranking algorithm for ICN contents to fight publication spam based on publisher and subscriber ranking. Ranking is based on the number of publications for publishers and the number of votes for subscribers. The authors assume many restrictions in ICN architecture such as anonymity of ICN users and the existence of authentication servers. Ghali et al. [39] present a ranking algorithm for ICN cached contents to mitigate content poisoning in NDN architecture. Ranking is based on user behaviors after receiving content objects. Our ranking calculates cached content popularities based on the weights of user requests and the ratio of requesting contents. Our solution is implemented in ICN routers and it does not require modifications to ICN architectures.

In the literature of DDoS attacks in general, there are many classifications for DDoS attacks and detection/prevention mechanisms [25, 47, 60, 90, 91, 94]. In current Internet architectures, most referenced countermeasures for DDoS are IP trace back, packet filtering, and rate limiting [50, 52, 57, 61, 86]. Also, there are other countermeasures for cache poisoning attacks as in DNSSEC and S-DNS [74]. Unfortunately, these techniques cannot be used in ICN architectures, because these techniques depend on IP addresses.

Also, there are other types of attacks for ICN architectures related to unauthorized access and privacy violations that may cause DDoS partially. For unauthorized access attacks, the solutions can be classified to centralized, decentralized, and encryption-based access control schemes [11, 35, 38, 54, 62, 70, 72, 80, 82]. Privacy related attacks include monitoring certain content names and timing analysis attacks [13, 21, 45, 63, 64].

The proposed solution presented in this chapter is designed specifically for ICN architectures, as it depends on unique ICN characteristics. The solution does not include any host addresses, which means that the solution does not depend on any IP-based addressing as in non-ICN architectures. The solution depends on in-network caching, which is one of the major ICN components that is not available in non-ICN architectures. The solution includes RSR metric, which depends on the ICN property that each request has one response and there is no response without a request. In non-ICN architectures, requests can receive many data packets.

5.3 DDoS Attacks

In this section, we classify and discuss DDoS attacks in ICN routing and caching and then demonstrate the different attack scenarios. This classification discusses DDoS attacks from attacker's action perspective. An attacker may cause:

- Denial of service by publishing malicious contents or announcing invalid routes for ICN contents.
- Denial of service by increasing the request timeout for some ICN nodes to violate the consistency between ICN asynchronous publication and the subscription processes.

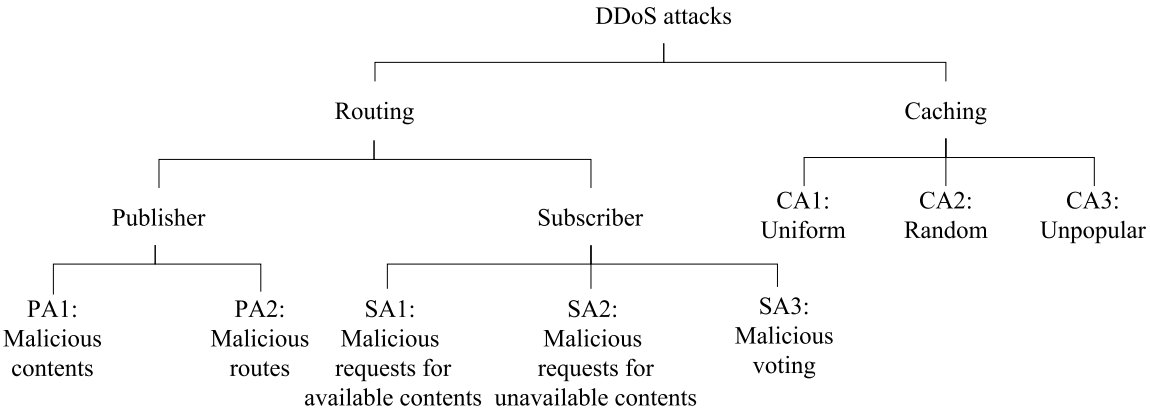

Figure 5.1: DDoS attacks in ICN. Publisher Attacks (PA1 and PA2), Subscriber Attacks (SA1, SA2, and SA3), and Caching Attacks (CA1, CA2, and CA3). Remainder of the chapter will refer to these attacks using these symbols

- Cache pollution by forcing ICN caching to store unpopular contents, and consequently satisfying all requests from the original sources rather than the closest available copies.

5.3.1 Classification of DDoS Attacks in ICN

DDoS attacks in ICN can be classified into routing and caching attacks, as shown in Figure 5.1. Routing attacks can be classified into publisher and subscriber DDoS attacks. In the caching attacks, an attacker forces the caching system to evict the most popular contents and store the least popular ones. The attacker sends different distributions to affect the caching system, which can be classified into uniform, random and unpopular distributions.

5.3.2 Routing Attacks

In the publisher attacks, an attacker publishes malicious contents or announces invalid routes. In the subscriber attacks, an attacker sends malicious requests for available and/or unavailable contents. An attacker can also send malicious votes against valid content to

decrease content's rating. Additionally, the attacker can do all the aforementioned attacks simultaneously. As shown in Figure 5.2, an attacker A_1 sends invalid contents or announces malicious routes for ICN content to cause DDoS and attract legitimate requests. An attacker A_2 who controls many end systems, sends a large number of malicious requests to an ICN to exhaust ICN resources such as memory and processing power. An attacker aims to fill ICN interest tables to cause DDoS for legitimate users. These malicious requests can be sent for available and unavailable contents. The attacked routers try to satisfy these malicious requests and forward them to neighbouring routers, which in turn forward these malicious requests to their neighbouring routers. In this case, legitimate requests take longer response times to be satisfied. If response time exceeds a certain threshold, legitimate requests will not be satisfied. The impact of this attack is amplified in ICN, because users retransmit unsatisfied requests, which add extra overload to ICN. This scenario may lead to denial of service or at least long delays. In this chapter, we address both malicious scenarios for publication and subscription.

5.3.3 Caching Attacks

An attacker sends uniform, random, or unpopular requests to spoil ICN caching by changing content popularities. These malicious requests force a caching system to store least popular contents and evict the popular ones. As depicted in Figure 5.2, normally when a user U_1 requests for a content for the first time, ICN responds with the content from the original source. If another user U_2 makes a request for the same content, the second user gets it from the closest available copy in router R_1 instead of the original source. If an attacker A_2 succeeds to pollute ICN caching, and a second user U_2 requests for the same content, the second user gets the content from the original data source instead of the closest

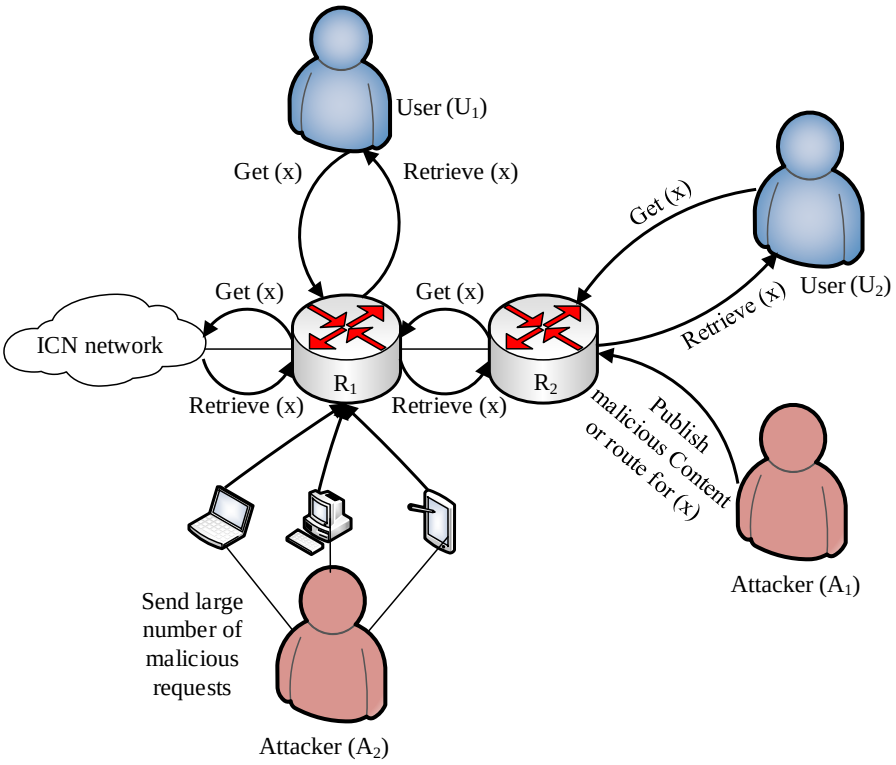

Figure 5.2: ICN routing and caching related attacks: Attacker A_1 publishes malicious content and/or route. Attacker A_2 sends a large number of malicious requests to overload routing and caching in ICN. U_1 and U_2 are legitimate users trying to get content x. R_1 and R_2 are ICN router with caching capabilities

available copy. The second user's request takes the full path as the first user's request in the attack case.

Time analysis attack

In ICN, in-network caching enables any node to cache contents. An attacker can send requests for cached and uncached contents and measure time differences between the two cases. This time difference can be used to detect if a nearby user has requested a certain content before or not. As depicted in Figure 3.8, the time required to send a request and

receive a response between the original data source and user's or attacker's edge router R_2 is T_1. Time required to send a request and receive a response between user U_2 or attacker A_2 and edge router R_2 is T_2. In a normal case, total time T_1+T_2 refers to round trip time from user U_2 or attacker A_2 to the original data source. In an attack case, if the attacker A_2 receives a response in only time T_2, he/she can use this time difference to violate ICN privacy for both contents and users. Attacker A_2 can know private information about content popularities and whether a proximate user has requested this content before or not. This attack is one of the caching related attacks that causes privacy violation. In this chapter, we are concerned only with this attack, not with other ICN privacy issues related to ICN contents or users.

5.3.4 Attack Scenarios

It is commonly agreed that web traffic follows Zipf-like distribution [17]. Other research shows that the Internet follows stretched exponential (SE) distribution [41]. Whichever the distribution is, an attacker performs the aforementioned attacks by changing request rate and request pattern (distribution) with respect to legitimate users. Table 5.1 shows the different scenarios and combinations of routing and caching attacks. When an attacker sends malicious requests with much higher request rate with the legitimate request pattern, this leads to routing attacks. Malicious requests can be sent for both available and unavailable contents. A caching attack results when an attacker sends malicious requests with different patterns with the legitimate request rate. Malicious request patterns can be uniform, random, and unpopular. In uniform distribution, an attacker sends the same number of malicious requests for all contents in a data set. In random distribution, an attacker sends

a random number of malicious requests for contents in a data set. In unpopular distribution, an attacker sends the same number of requests for just unpopular contents in a data set. When an attacker changes request rate and request pattern, the attacker performs both attacks simultaneously.

Table 5.1: Attack methodology (LB: Legitimate behavior, AB: Attack behavior)

		Request rate	
		LB	**AB**
Request pattern	**LB**	Legitimate behavior	Routing attacks
	AB	Caching attacks	Rotuing and Caching attacks

5.4 Proposed Countermeasures

In this section, we present our proposed solution for DDoS attacks in ICN routing and caching (DADI). We start by the methodology for calculating the threshold values. We then show the detection and prevention parameters. We present the algorithms in the last subsection. Our goal is to protect the ICN against DDoS attacks, achieving the following explicit design goals:

- Availability maintenance: We design our solution to maintain the network availability for legitimate users by detecting and preventing ICN DDoS attacks coming from publishers and/or subscribers.

- Caching protection: We propose a solution that preserves the ICN universal caching feature and maintains the most popular contents in order to deliver ICN contents efficiently.

- ICN utilization: The solution increases the ICN utilization for legitimate users in

terms of routing and caching. We use two performance metrics to measure the improvement in ICN utilization based on our solution.

The proposed solution consists of routing and caching countermeasures, as shown in Figure 5.3. The proposed solution is implemented in ICN routers and effectively detects and prevents malicious requests in ICN caching and routing based on threshold values as described in the following subsections. The request rate, rating for contents, rating for publishers, and rating for cached content parameters are dependent on the RSR. RSR represents the number of satisfied requests per user with respect to the outgoing requests from this user. RSR depends on an ICN property that each request has only one response and there is no response without a request [3]. In non-ICN architectures, a request can receive many data packets. Request rate indicates the number of outgoing requests per second for each interface. Cache hit ratio is the number of cache hits per user with respect to the outgoing requests from this user. Rating for contents indicates the ranking method for ICN contents to select the best available content. Rating for publishers represents the ranking technique for ICN publishers to select the most trusted publisher.

The RSR, request rate, rating for contents, and rating for publishers are used to handle the attack scenarios, when the subscriber sends malicious requests for available and/or unavailable contents (SA1 and SA2). The ICN router sends a test message to address the attack, when the publisher announces malicious routes (PA2). The RSR, rating for contents, and rating for publishers are used to handle the attacks, when the publisher publishes malicious contents or subscriber sends malicious votes (PA1, PA2, and SA3). The RSR and rating for cached contents are used to address DDoS effects on the caching (CA1, CA2, and CA3).

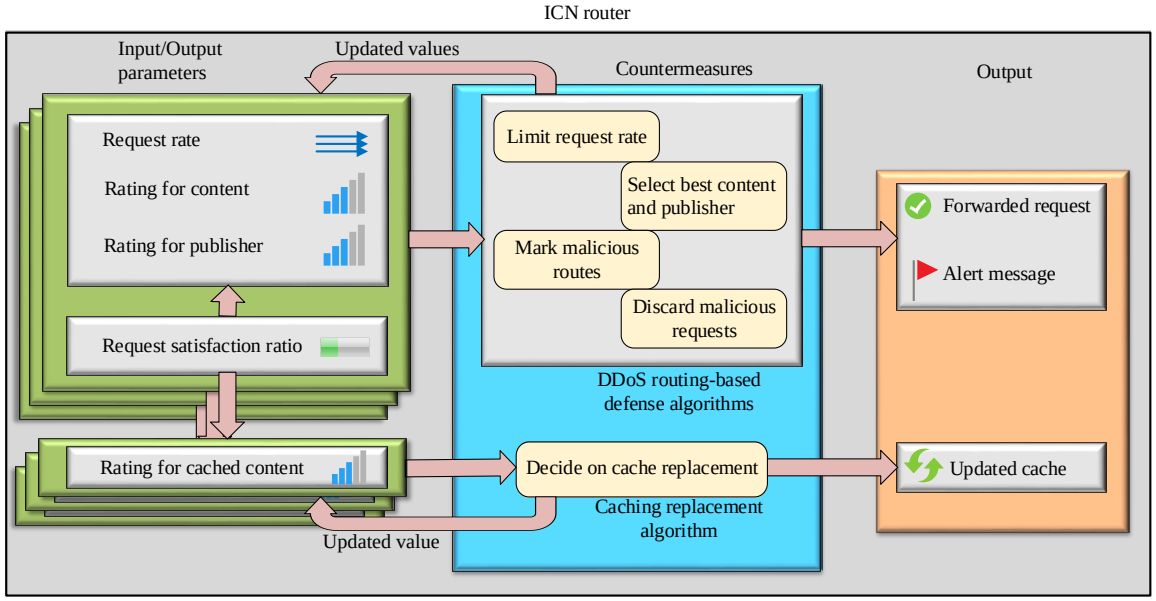

Figure 5.3: Proposed countermeasures and parameters

5.4.1 Threshold Value Calculation

In calculating the threshold values, we assume a period with no malicious requests for calculations. We study the normal behavior of legitimate users and calculate threshold values for the following parameters:

- Request Satisfaction Ratio (RSR): Request satisfaction ratio for user i connected to router k via interface j is calculated by the following equation:

$$RSR_{i,j,k} = \frac{Number\ of\ satisfied\ requests}{Number\ of\ outgoing\ requests} \tag{5.1}$$

Then we calculate $RSR_{threshold}$ for legitimate users, which refers to minimum legitimate RSR value.

- Request Rate: The request rate for user i connected to router k via interface j is

calculated by the following equation:

$$RequestRate_{i,j,k} = \frac{Number\ of\ outgoing\ requests}{\text{Sec}} \tag{5.2}$$

Then we calculate $RequestRate_{threshold}$ for legitimate users, which refers to maximum legitimate request rate value.

- Cache Hit Ratio: The cache hit ratio for user i connected to router k via interface j is calculated by the following equation:

$$CacheHitRatio_{i,j,k} = \frac{Number\ of\ cache\ hits}{Number\ of\ outgoing\ requests} \tag{5.3}$$

Then we calculate $CacheHitRatio_{thresholdU}$ and $CacheHitRatio_{thresholdL}$ for legitimate users, which refers to maximum and minimum legitimate cache hit ratio.

- Ratings for contents and publishers: The equations for content and publisher ratings are mentioned in the prevention phase. Then we calculate the maximum threshold values for $ContentRating_{threshold}$ and $PublisherRating_{threshold}$.

These threshold values are calculated using the following procedure. For each time slot (sample) in a dataset, each ICN router records RSR, request rate, and cache hit ratio of different users. For each sample, each ICN router calculates the minimum RSR, maximum request rate, and maximum and minimum for cache hit ratio. For all samples, each ICN router calculates mean $(\bar{x})$ and standard deviation (s) for minimums and maximums for each metric separately. Threshold values are calculated as follows. In case of maximum threshold value, we add mean $\bar{x}$ to a constant multiplied by standard deviation. In case of minimum threshold value, we subtract the standard deviation s multiplied by constant from

the calculated mean $\bar{x}$. The value of constant *(c_1, c_2, c_3, c_4, and c_5)* multiplied by standard deviation detects how far our threshold value is from the calculated mean. The threshold value for RSR is:

$$RSR_{threshold} = \bar{x}_{RSR} - c_1 * s_{RSR} \tag{5.4}$$

The threshold value for request rate is:

$$RequestRate_{threshold} = \bar{x}_{RequestRate} + c_2 * s_{RequestRate} \tag{5.5}$$

The threshold values for cache hit ratio is:

$$CacheHitRatio_{thresholdU,thresholdL} = \bar{x}_{CacheHitRatio} \pm c_3 * s_{CacheHitRatio} \tag{5.6}$$

The above equation calculates maximum and minimum threshold values for cache hit ratios. Then ICN routers send the new values of the requested user to the storage locations to update the content and publisher ratings. The threshold value for rating for contents is:

$$ContentRating_{threshold} = \bar{x}_{ContentRating} + c_4 * s_{ContentRating} \tag{5.7}$$

The threshold value for rating for publishers is:

$$PublisherRating_{threshold} = \bar{x}_{PublisherRating} + c_5 * s_{PublisherRating} \tag{5.8}$$

The use of more than one threshold value allows the detection and prevention phases to effectively define attacker behaviors. These values monitor attacker behaviors from different perspectives and compare it to legitimate behaviors. An attacker has to reduce his or

her request rate and request pattern to follow the legitimate bounds of the threshold values, and this makes the attacks futile.

5.4.2 Detection Parameters

In the detection phase, each ICN router is able to differentiate between legitimate and attack behaviors and detect which attack case is happening to take the appropriate actions in the prevention phase. A request rate of a user above $RequestRate_{threshold}$ is an indication of an attacker sends a large number of requests either for available or unavailable contents. A request rate of a user below $RequestRate_{threshold}$, while the RSR of this user is below $RSR_{threshold}$, indicates an attacker sends a large number of requests for unavailable contents. When the request rate of a user is below $RequestRate_{threshold}$, while the cache hit ratio of this user is below $CacheHitRatio_{thresholdL}$ value means an attacker tries to exhaust and overload ICN architecture and pollutes ICN caching to store unpopular contents. In a final case, a cache hit ratio of an interface is above $CacheHitRatio_{thresholdU}$ means an attacker has violated ICN privacy by monitoring his/her proximate users. For an attacker to pass these conditions, he/she needs to send malicious requests within legitimate rates, which minimize the impacts of the performed attacks.

Before a user receives the requested content, the ICN router checks if the rating for this content or the publisher is above $ContentRating_{threshold}$ and $PublisherRating_{threshold}$ values. The ICN router then sends an alert message to notify the user, who decides whether to accept or reject this content. An ICN router sends an alert to a subscriber when a content or publisher rate is more than a certain threshold value. For an attacker to pass these conditions, the attacker needs to send malicious requests within legitimate rates that minimize the impact of the performed attacks.

5.4.3 Prevention Parameters

This subsection describes the countermeasure parameters for the proposed solution including the request rate limit, rating for contents, rating for publishers, test message and rating for cached contents. We refer to user i connected to router k via interface j by $U_{i,j,k}$. Only edge routers depend on the connected users. Other routers calculate the values for each interface based on the neighboring routers without the knowledge of end-users. After the edge routers, the same equations will be used only for the connected routers.

Request rate limit: By using the threshold values, the prevention phase is able to protect ICN against the combinations of malicious requests for either available or unavailable contents in ICN routing and caching and take the appropriate actions. When an attacker sends malicious requests for available and unavailable requests, each ICN router limits the incoming requests from user i using the following equation:

$$RequestRateLimit_{i,j,k} = \begin{cases} RSR_{i,j,k} * RequestRate_{threshold}, & \text{if } RequestRate_{i,j,k} > RequestRate_{threshold} \\ & \\ RSR_{i,j,k} * RequestRate_{i,j,k}, & \text{if } RequestRate_{i,j,k} < RequestRate_{threshold} \\ & \text{AND } (RSR_{i,j,k} < RSR_{threshold} \text{ OR} \\ & CacheHitRatio_{i,j,k} < CacheHitRatio_{\text{thresholdL}}) \end{cases} \tag{5.9}$$

Rating for contents: This parameter ranks ICN contents to enable an ICN router to select the best available content. The voting weight against content c ($W_{content}$) is calculated by

the following equation:

$$W_{content} = \sum_{v=1}^{n} \frac{CV_{U_{i,j,k}}}{Number\ of\ U_{i,j,k}\ votes} * RSR_{i,j,k} \tag{5.10}$$

where $CV_{U_{i,j,k}}$ is the $U_{i,j,k}$ votes *(v)* against content *c*, and *n* is the number of votes against this content. The voting ratio against content *c* ($R_{content}$) is calculated based on the following equation:

$$R_{content} = \frac{Number\ of\ votes\ against\ c}{Number\ of\ downloads\ for\ c} \tag{5.11}$$

From Equations (5.10) and (5.11), we derive the following equation:

$$Rating\ for\ content\ c = W_{content} * R_{content} \tag{5.12}$$

Rating for publishers: This parameter ranks ICN publishers. As a result, an ICN router can select the best trusted publisher. The voting weight against publisher *p* ($W_{publisher}$) is calculated by the following equation:

$$W_{publisher} = \sum_{v=1}^{n} \frac{PV_{U_{i,j,k}}}{Number\ of\ U_{i,j,k}\ votes} * RSR_{i,j,k} \tag{5.13}$$

where $PV_{U_{i,j,k}}$ is the $U_{i,j,k}$ votes *(v)* against publisher *p*, and *n* is the number of votes against this publisher. The voting ratio against publisher *p* ($R_{publisher}$) is calculated by the following equation:

$$R_{publisher} = \frac{Number\ of\ publications\ that\ received\ voting}{Number\ of\ publications\ from\ p} \tag{5.14}$$

From Equations (5.13) and (5.14), we get the following equation:

$$Rating\ for\ publisher\ p = W_{publisher} * R_{publisher} \tag{5.15}$$

Test message: ICN router sends a test message through a route that does not return any response to check whether this route is malicious or not.

Rating for cached contents: In the cache pollution case, if there is a cache hit at an ICN router, the ICN router returns content and updates cache hit ratio for the requested user. In the case of cache miss, an ICN router only caches content if the following conditions hold, which indicate legitimate behaviors. Otherwise, ICN routers do not cache such content.

$$\begin{aligned} RSR_{i,j,k} &> RSR_{threshold}\ \text{AND} \\ RequestRate_{i,j,k} &< RequestRate_{threshold}\ \ \text{AND} \\ CacheHitRatio_{i,j,k} &> CacheHitRatio_{\text{thresholdL}} \end{aligned} \tag{5.16}$$

This parameter for cache pollution attacks ranks ICN cached contents at each router. ICN caching evicts least popular contents. As the number of user requests increases, the user request weight decreases. The rating depends on RSR, number of user requests, number of requesting users and total number of users. The request weight for cached content c $(W_{cachedcontent})$ is calculated by the following equation:

$$W_{cachedcontent} = \sum_{r=1}^{n} \frac{R_{U_{i,j,k}}}{Number\ of\ U_{i,j,k}\ requests} * RSR_{i,j,k} \tag{5.17}$$

where $R_{U_{i,j,k}}$ is the $U_{i,j,k}$ requests (r) for cached content c, and n is the number of requests.

The cache request ratio for content *c* ($R_{cachedcontent}$) is calculated by the following equation:

$$R_{cachedcontent} = \frac{Number\ of\ requesting\ users}{Number\ of\ users} \tag{5.18}$$

From Equations (5.17) and (5.18), the rating for cached content popularity (c) is calculated as

$$Rating\ for\ cached\ content\ c = W_{cachedcontent} * R_{cachedcontent} \tag{5.19}$$

5.4.4 Algorithms

Algorithm 1 presents the proposed solution for the ICN routing related DDoS attacks. The algorithm uses two functions (update request rate limit and user voting) that are shown in Algorithm 2, and Algorithm 3, respectively. Algorithm 4 presents the proposed solution for ICN caching related DDoS attacks. Algorithm 5 and Algorithm 6 show the proposed solution for time analysis attack.

Algorithm 1 is implemented in ICN routers. Once a subscriber sends a request to an ICN router, the router checks the request availability (Line 1). If this request passes the limit check, the router forwards it and waits for the response (assuming the requested content is not in the cache) (Lines 2 and 3).

ICN routers find the best available contents based on our ranking to ICN contents and publishers. ICN routers satisfy user requests by selecting the top-ranked ones, if there are multiple content copies available (Line 4). The ICN router checks the response in three cases. First, when the content is found (Line 5), the router updates the RSR for this user (Line 6) and checks the rating for ICN contents and publishers (Line 7). If the rating is more than the threshold values, the ICN router sends an alert message to the subscriber

Algorithm 1 Solution steps and countermeasures for ICN routing related DDoS attacks scenarios. The code in the comments indicates the attack types (see Figure 5.1)

Input: Incoming request via user $U_{i,j,k}$

1: **if** $RequestRateLimit_{i,j,k} < RequestRate_{threshold}$ **then**
2: Add request to the pending interest table
3: Forward the request
4: Get the best available content // PA1 and PA2
5: **if** *ReturnStatus* is *Content found* **then**
6: Update $RSR_{i,j,k}$
7: **if** $ContentRating > ContentRating_{threshold}$ OR $PublisherRating > PublisherRating_{threshold}$ **then**
8: Send an alert message
9: Get AlertResult
10: **end if**
11: **if** *AlertResult* is *Yes* **then**
12: Send the content back to the user
13: Call *UserVoting* Algorithm // PA1, PA2, and SA3
14: **end if**
15: **else**
16: **if** *ReturnStatus* is *Nocontent found* **then**
17: Update $RSR_{i,j,k}$
18: **else** // Request timeout
19: **if** $RSR_{i,j,k} < RSR_{threshold}$ **then**
20: Update $RSR_{i,j,k}$
21: **else**
22: Send a test message
23: **if** *ReturnStatus* is *Content found* **then**
24: Resend the request
25: **else**
26: Mark this route as invalid route // PA1
27: Update $RSR_{i,j,k}$
28: **end if**
29: **end if**
30: **end if**
31: **end if**
32: Call $UpdateRequestRateLimit_{i,j,k}$ Algorithm // SA1 and SA2
33: **else**
34: Discard the request // SA1 and SA2
35: **end if**

(Line 8), who decides whether to accept this content or not. If the subscriber accepts the content, the alert result will be set to yes, which is the default value. Otherwise, the alert result will be set to no (Line 9). If the subscriber receives a content, he/she can vote against it (Lines 10-14). Second, when there is no content found with the requested name, the router just updates the RSR for this user (Lines 15-17). Third, when the request is timed out, the router first checks the behavior of this user. If the RSR of this user is below a threshold, then the router directly updates the RSR for this user (Lines 18 -20). If the RSR is above the threshold value, the router sends a test message to check the announced route. If the router gets a response, then it retransmits the request again and restarts the whole process (Lines 21-24). Otherwise, it marks this route as a malicious one and also updates the RSR (Lines 25-28). The ICN router updates the request rate limit for this user (Line 32). If the request does not pass the limit check, the ICN router discards this request (Line 34).

In Algorithm 2, the ICN router implements the conditions and assignments mentioned in Equation 5.9.

Algorithm 2 Update request rate limit

Input: Incoming request via user $U_{i,j,k}$

1: **if** $RequestRate_{i,j,k} > RequestRate_{threshold}$ **then**
2: $\quad RequestRateLimit_{i,j,k} = RSR_{i,j,k} * RequestRate_{threshold}$
3: **else**
4: $\quad$ **if** $RSR_{i,j,k} < RSR_{threshold}$ OR $CacheHitRatio_{i,j,k} < CacheHitRatio_{thresholdL}$ **then**
5: $\quad\quad RequestRateLimit_{i,j,k} = RSR_{i,j,k} * RequestRate_{i,j,k}$
6: $\quad$ **end if**
7: **end if**

Algorithm 3 describes the user voting against a false content. ICN routers forward the voting message with the calculated weight to the storage location. ICN updates the ratings for this content and publisher according to Equations 5.12 and 5.15.

Algorithm 3 User voting

Input: Received content upon user's request

1: **if** *content* is *invalid* **then**
2: User $U_{i,j,k}$ votes against content c or publisher p
3: Router R_k calculates the associated weight for $U_{i,j,k}$ vote
4: Update ratings for content c and publisher p
5: **end if**

Algorithm 4 shows the caching policy and replacement strategy. If a legitimate cache request is received, then ICN caching updates the cached content rating using Equation 5.19 and inserts a new cache entry for this content (Lines 1, 2, and 6). If the cache is full, then ICN caching evicts the least popular and least recently used cached content (Lines 3-5). Otherwise, if the request is not a legitimate one, ICN router does not cache this content.

Algorithm 4 Solution steps and countermeasures for ICN caching related DDoS attacks scenarios

Input: Incoming request via user $U_{i,j,k}$

1: **if** $RSR_{i,j,k} > RSR_{threshold}$ AND
$RequestRate_{i,j,k} < RequestRate_{threshold}$ AND
$CacheHitRatio_{i,j,k} > CacheHitRatio_{thresholdL}$ **then**
2: Update the cached content rating
3: **if** *cache* is *full* **then**
4: Evict the least popular and LRU content
5: **end if**
6: Insert new cache entry
7: **else**
8: No caching
9: **end if**

In case of time analysis attack, Algorithm 5 shows our prevention technique for this attack. Each ICN router records round trip times for cached contents. In case of attack detection, the ICN edge router connected to the requested interface responds with random delays close to the original round trip times. Also, each ICN router sends an alert message to its attacked users. We use U_i to refer to an input user to the algorithm, who will be tested

to detect if he/she is an attacker or not. We use U_k to refer to a legitimate user connected to interface k. We check the attacked user list coming from Algorithm 6. Then we send an alert message to the attacked users to notify them that a nearby user is monitoring their requests.

Algorithm 5 Time Analysis Attack Prevention

Input: User (U_i) sends a request to ICN router

```
1: Update CacheHitRatio (U_i)
2: if CacheHitRatio (U_i) > CacheHitRatio_thresholdU then
3:     Mark U_i as an attacker
4:     Add random delays to U_i responses
5:     AttackedUserList = CheckAttackedUsers (U_i)
6:     if Count (AttackedUserList == 1) then
7:         Send notification message to U_k
8:     else
9:         if (1 <Count (AttackedUserList)<Max) then
10:            loop (AttackedUsersList)
11:                Send notification message for each user (U_k) in AttackedUserList
12:            end loop
13:        else
14:            Send broadcast notification message to all
15:        end if
16:    end if
17: else
18:    Handle request normally
19: end if
```

Algorithm 6 shows how we detect the attacked users by counting and grouping the common requests between an attacker and each proximate user. We use A_i to refer to an attacker connected to interface i. If the number of matched requests between an attacker A_i and a user U_k exceeds $CacheHitRatio_{thresholdL}$, then we add this user to the attacked user list. In order to prevent time analysis there are two techniques: using random delays close to original response time; generating cache misses for attacker requests. Both techniques have similar impacts from a consumer side, because he/she gets a response after approximately

the same time. We prefer the first technique to save network resources instead of getting contents from the original sources every time.

Algorithm 6 Check Attacked Users

Input: Attacker (A_i)

1: Get matched requests between A_i and each user U_k
2: TempAttackedUserList = count matched requests group by user U_k
3: **loop** (TempAttackedUserList)
4: **if** count(entry)/NoUsrReq(A_i)>$CacheHitRatio_{\text{thresholdL}}$ **then**
5: Add user U_k to AttackedUserList
6: **end if**
7: **end loop**
8: **return** AttackedUserList

5.4.5 DADI Analysis

This section analyses DADI in more detail. DADI also has the following characteristics:

Dynamic control of attacker behavior. Attackers can change their rates and patterns continuously. DADI uses different threshold values to force attackers to send requests within allowable limits. DADI takes the appropriate actions, if any violations of these limits happen. These threshold values can be statically set or dynamically calculated based on the application domain.

Subscriber-driven solution. Complying with ICN receiver-driven architecture, DADI defends against different DDoS attacks upon receiving attacker requests. DADI does not detect or prevent publisher attacks until a subscriber requests for these malicious routes or contents. Upon receiving a request for a malicious route, DADI checks the route and notifies the subscriber. For the malicious contents, the subscriber votes against it and DADI decreases content ratings.

Subscriber involvement. DADI incorporates subscribers to be a part of the solution. Subscribers can accept or reject contents based on their ratings. DADI sends an alert message to attacked subscribers, when there is a nearby attacker trying to monitor subscribers requests.

Defending against complex attack scenarios. An attacker can create complex attacks that combine more than one type at the same time. For example, an attacker sends a large number of unpopular requests for available contents to cause routing and caching effects simultaneously. Using different countermeasures for each attack type and case, DADI is able to defend against these complex scenarios.

Defending against collaborating attackers. Attackers as publishers and subscribers can work together to cause maximum damage to ICN architecture. For example, an attacker publishes invalid contents and malicious subscribers request for this content and do not vote against it. This can be done on a large scale to cause DDoS and increases invalid content ratings. Using rate limits and changing voting weights as the number of subscriber vote increase, DADI presents solutions for these scenarios.

Limitations. The worst case scenario for DADI is when an attacker sends malicious requests up to the upper limits of the threshold values and then behaves legitimately until the attacker enhances his recorded values in attached ICN routers. Afterwards, the attacker starts to send malicious requests up to the upper limits again and so on. The attacker needs to have a prior knowledge of the legitimate limits and adjust the sent requests to balance his recorded values. This scenario is considered to be most harmful in DADI, and can still cause some damage within the legitimate user bounds.

The worst case scenario for DADI in caching attacks is when an attacker sends uniform requests, DADI detects and prevents random and unpopular requests easier than uniform requests. The attacker tries to increase the cache hit ratio and RSR by sending uniform

requests.

DADI focuses on DDoS related attacks in ICN routing and caching. There are other types not covered by DADI that can also cause DDoS such as blocking certain contents or requests in naming attacks.

5.5 Experimental Evaluation

In this section, we evaluate the impact of routing and caching DDoS attacks with and without our proposed countermeasures. We use real data trace for web traffic, which contains 1.3 billion web requests recorded at servers for the 1998 World Cup [102]. The simulation parameters are detected based on the real data trace as follows: request rate for legitimate users is 20 request per second, users send 80% new requests and 20% non-modified requests, top 10% of files receive about 90% of requests, about 10% of contents are requested only one time. We set the attacker request rate to be 100 requests per second. We also use the following parameters: no. of Pending Interest Table (PIT) entries = 1000 entry, payload size = 1 KB, no. of cache entries = 1000 entry, request expiration timeout = 4 seconds. During the experiments, we start implementing the solution after 10% of simulation time to give users the opportunity to send their requests and then apply the proposed countermeasures. In all caching results, we record the results starting from the fifth second in order to give an opportunity to ICN users to send their requests and build their cache hit ratios before comparing ICN user behaviors to the threshold values.

For security analysis of the proposed rating equations mentioned in subsection 5.4.3, by analyzing the rating equations for object *x* in the attack case. Assuming that all attackers have the same trust value *(TV)* and each one sends one request against object x, the rating

becomes as follows:

$$Rating\ for\ object\ x = \frac{M^2 * TV}{m * (M+N)} \tag{5.20}$$

where M is the number of the attackers, N is the number of the legitimate users, m is the number of the requests per attacker, and TV is between 0 and 1; 0 when all requests are not satisfied and 1 when all requests are satisfied. The preceding equation shows that, in order to perform these attacks, the attacker needs to be distributed in many locations and votes against a limited number of contents. As the attacker targets more objects from limited locations, his/her voting weight and trust value decline. The proposed solution in this chapter makes these attack scenarios difficult and lessens its impact on the ICN.

We compare our results with the leading solutions in each category. For publisher side DDoS attacks, our results are compared with the needle in a haystack solution [39]. We compare subscriber side routing related DDoS attacks with the satisfaction-based push-back mechanism [8]. Caching related DDoS attacks are compared with respect to the CacheShield scheme [84]. These solutions achieve results close to the baseline in some attack scenarios.

5.5.1 Routing Attack Experiments

The main objective of an attacker in these types of attacks is to decrease the number of satisfied legitimate requests. Hence, we use satisfied legitimate request metric as an indicator of improvement achieved using our solution in the existence of routing attacks. Results are obtained for different ratios of attackers to legitimate users (20%, 50%, and 80%). Figure 5.4 - Figure 5.9 compare the satisfied legitimate requests for the following cases: baseline case; in existence of attacks; in existence of the attacks using a competitive solution; and in existence of attacks with our solution. The results show how our solution

enhances satisfied legitimate requests compared with the baseline case. According to the real data trace, the ratio for satisfied legitimate requests is about 99.5% when there are no attacks (baseline).

Figure 5.4 - Figure 5.6 show the results for malicious contents and malicious routes from a publisher side and malicious voting from a subscriber side (PA1, PA2, and SA3). The satisfied legitimate requests on average are 60%, 21%, and 6%, when 20%, 50%, and 80% attackers exist, respectively. Using our solution, the satisfied legitimate requests in the existence of these attacks are enhanced to about 93% in the worst case and 95% on average. In needle in a haystack solution, when the number of attackers and malicious contents, routes, and voting increases the satisfied legitimate requests gradually decreases. The satisfied legitimate requests is about 85% in the worst case and 91% on average. In this technique, the ranking algorithm depends on three main parameters: number of exclusions, time distribution of exclusions, and excluding interfaces ratio. The primary reason for this decrease is the needle in a haystack technique does not take into consideration the continuous malicious behavior of attackers by sending invalid contents and voting against valid ones.

Figure 5.7 - Figure 5.9 show the results for malicious requests for available and/or unavailable contents from a subscriber side (SA1 and SA2). When 20% attackers exist, satisfied legitimate requests decrease to about 55%. There is a reduction in the satisfied legitimate requests to about 25%, when 50% of users are attackers. The satisfied legitimate requests decline to about 10%, when 80% attackers exist. Based on our solution, satisfied legitimate requests in the existence of these attacks in the worst case is about 91%. Satisfaction-based pushback mechanism achieves the same results when all the malicious requests are for unavailable contents (RSR = 0). As the number of malicious requests

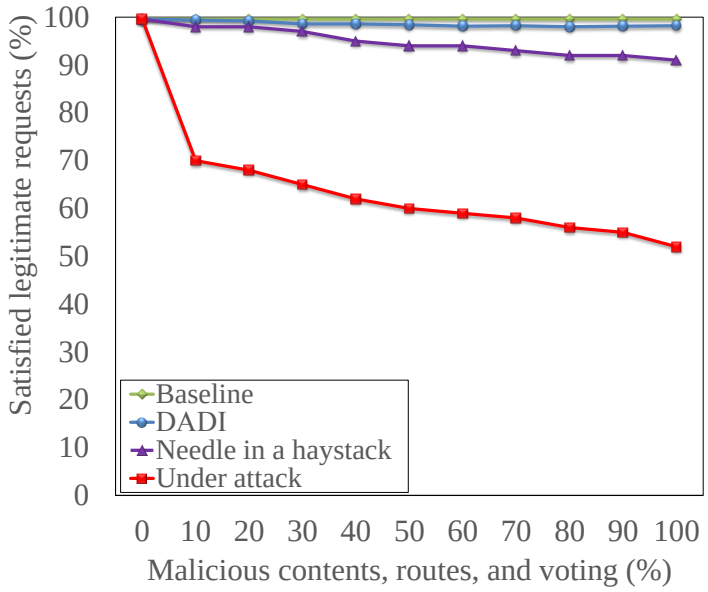

Figure 5.4: Percentage of satisfied legitimate requests (Routing attacks: PA1, PA2, and SA3), when 20% attackers exist

for available contents increases, the satisfied legitimate requests using satisfaction-based pushback mechanism rapidly decreases. The reason is that the satisfaction-based pushback mainly depends on one metric, which is the RSR.

5.5.2 Caching Attack Experiments

An attacker's main objective in these types of attacks is to decrease cache hit ratio to force ICN to retrieve data from the original source every time. We use cache hit ratio as a metric of improvement achieved by our solution in the existence of caching attacks. Results are also obtained for different ratios of attackers to legitimate users (20%, 50%, and 80%). We consider different distribution patterns, namely uniform, random, and unpopular. The results are in Figure 5.10 - Figure 5.12. Each figure compares cache hit ratio among the following cases: baseline case; in existence of attacks; in existence of the attacks using a CacheShield solution; and in existence of attacks with our solution. The results show

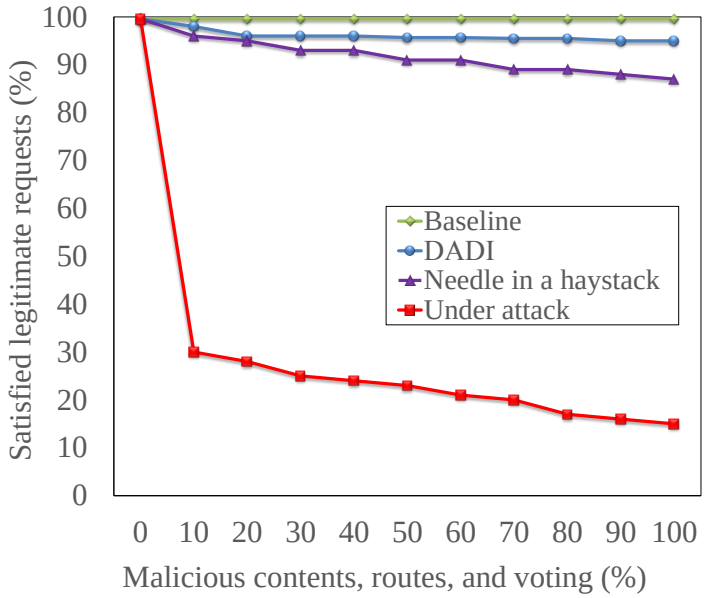

Figure 5.5: Percentage of satisfied legitimate requests (Routing attacks: PA1, PA2, and SA3), when 50% attackers exist

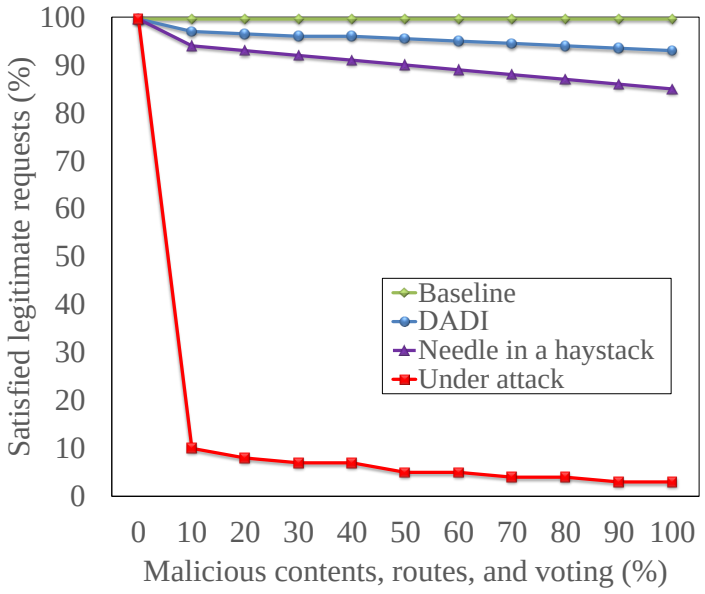

Figure 5.6: Percentage of satisfied legitimate requests (Routing attacks: PA1, PA2, and SA3), when 80% attackers exist

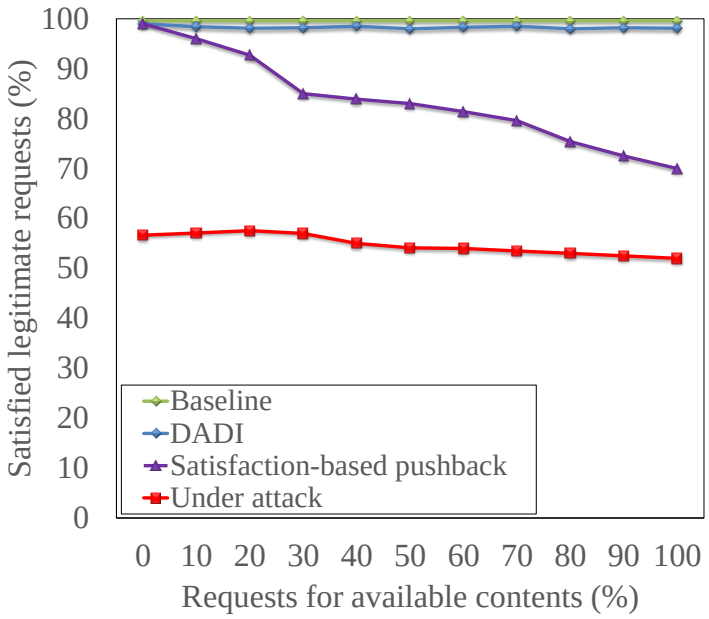

Figure 5.7: Percentage of satisfied legitimate requests (Routing attacks: SA1 and SA2), when 20% attackers exist

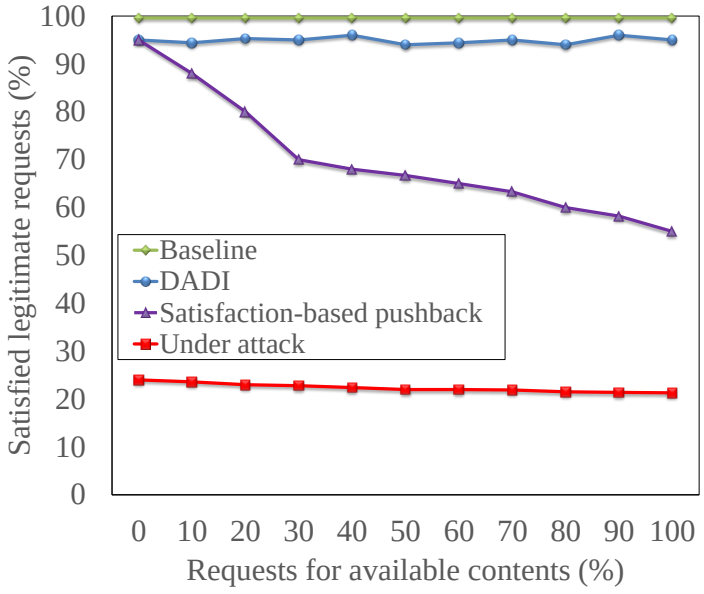

Figure 5.8: Percentage of satisfied legitimate requests (Routing attacks: SA1 and SA2), when 50% attackers exist

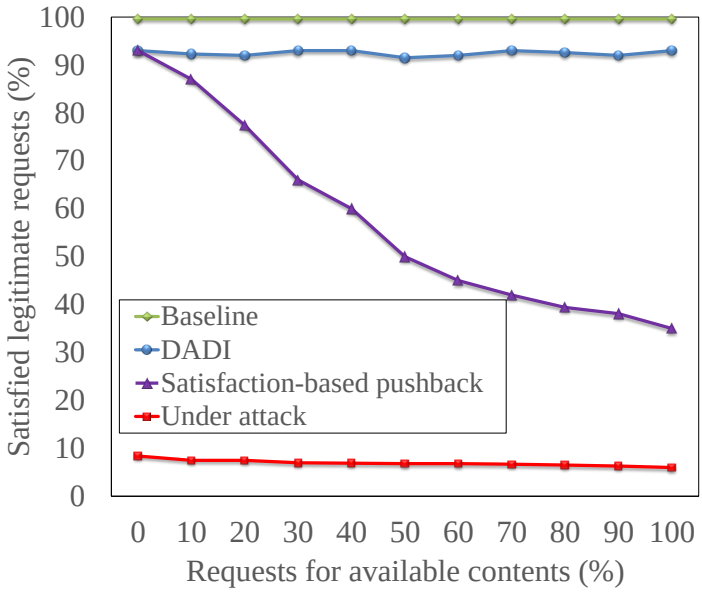

Figure 5.9: Percentage of satisfied legitimate requests (Routing attacks: SA1 and SA2), when 80% attackers exist

how the solution enhances cache hit ratio and generates results above the baseline case. According to the real data trace, the ratio for cache hits is about 39% when there is no attack (baseline).

Uniform distribution. When 20%, 50%, and 80% attackers exist, cache hit ratio decreases to about 18%, 13%, and 10%, respectively. Based on our solution, cache hit ratio in the existence of these attacks is about 51%, which enhances cache hit ratio over the baseline. CacheShield enhances the cache hit ratio to about 45% in the existence of these attacks.

Random distribution. There is a reduction in the cache hit ratio to about 19%, 16%, and 15% when there are 20%, 50%, and 80% attackers exist, respectively. The cache hit ratio is enhanced using our solution and in the existence of these attacks to about 55%, which also enhances cache hit ratio over the baseline. The cache hit ratio is increased to 49% using CacheShield algorithm.

Unpopular distribution. Without our solution and in the existence of the attacks, cache hit

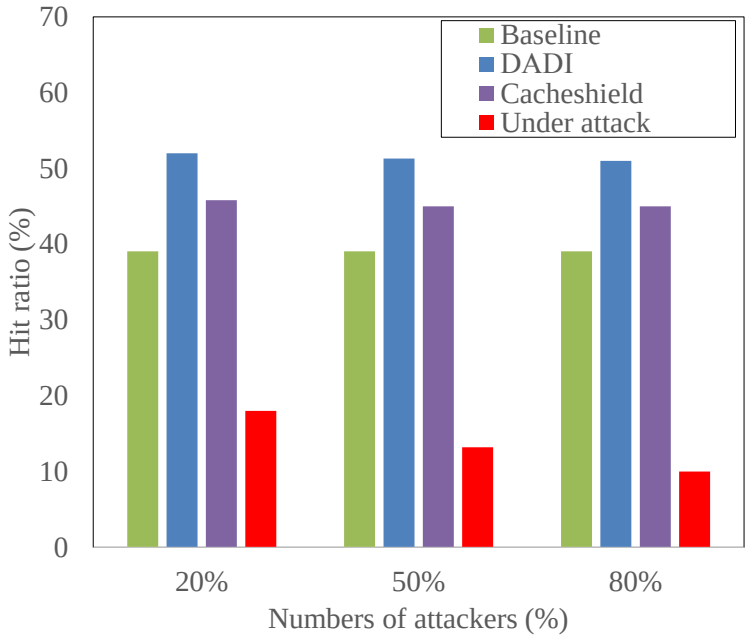

Figure 5.10: Percentage of cache hit ratio under caching attacks (Caching attacks: CA1)

ratios dramatically decline to about 14%, 10%, and 9% when 20%, 50%, and 80% attackers exist, respectively. With our solution and in the existence of these attacks, the cache hit ratio is improved to about 61%, which is also better than the baseline. CacheShield achieves about 64% cache hit ratio in the existence of these attacks that precedes our solution in this case. This improvement is because CachShield decides what to cache in a more effective manner than ours using its shield function in the case of unpopular requests. It is easier for an attacker to send uniform requests for all contents than just for unpopular contents.

Time analysis experiments

We measure cache hit ratio for attackers with and without the solution. In time analysis attack, we perform our experiments using only 10% attackers. The main objective for an attacker is to measure the time difference between cached and uncached contents. We are interested in caching impacts instead of the number of attacker requests. Within the time

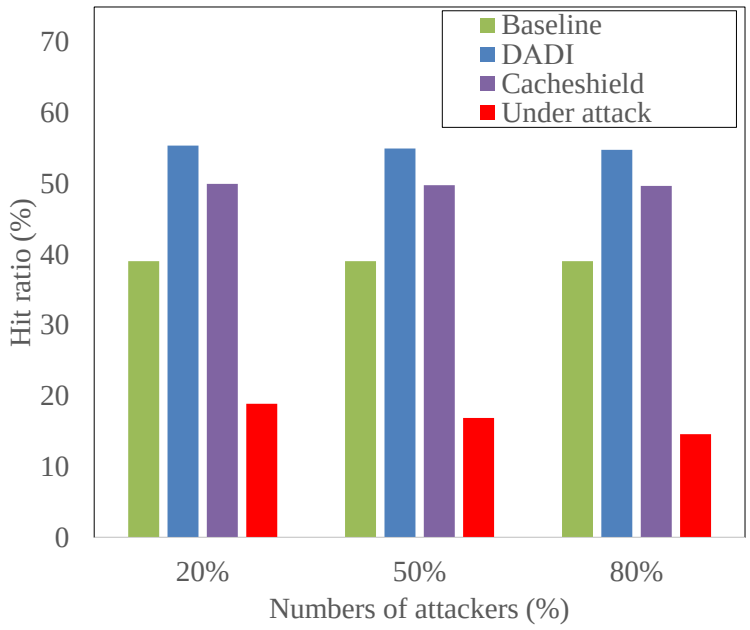

Figure 5.11: Percentage of cache hit ratio under caching attacks (Caching attacks: CA2)

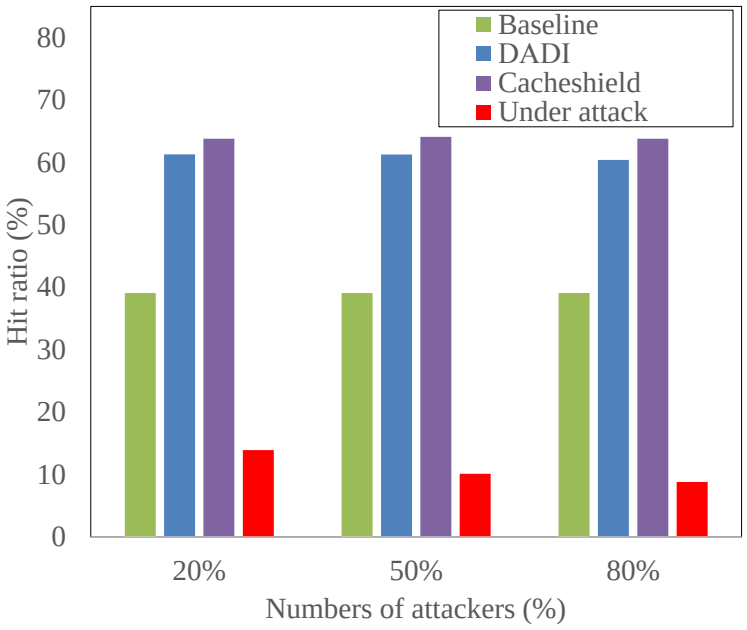

Figure 5.12: Percentage of cache hit ratio under caching attacks (Caching attacks: CA3)

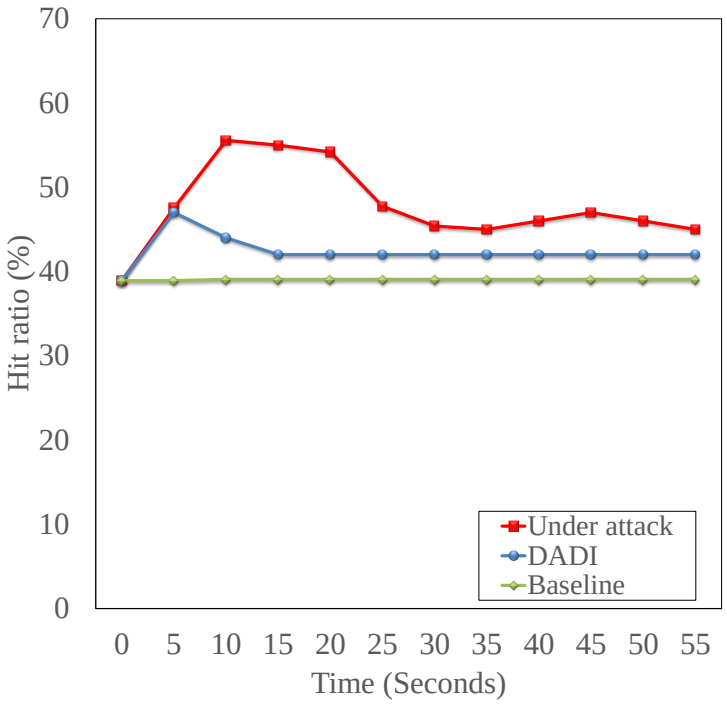

Figure 5.13: Percentage of cache hit ratio under time analysis attacks

expiration period, the attacker sends his/her requests in a continuous manner to monitor the neighbouring users. As shown in Figure 5.13, when 10% attackers exist and send specified requests for certain contents, cache hit ratio increases to about 47%. Based on our solution, cache hit ratio in the existence of these attacks is about 42%, which decreases attacker hit ratios. Figure 5.14 shows the percentage of positive likelihood ratio with and without the solution under time analysis attacks.

5.5.3 Discussion

As observed in the experimental results, our solution enhances ICN performance in routing and caching attack cases. With respect to related work [8, 27, 28, 34, 37, 39, 84], our solution achieves better or similar results. In [84], the results for cache hit ratio are better

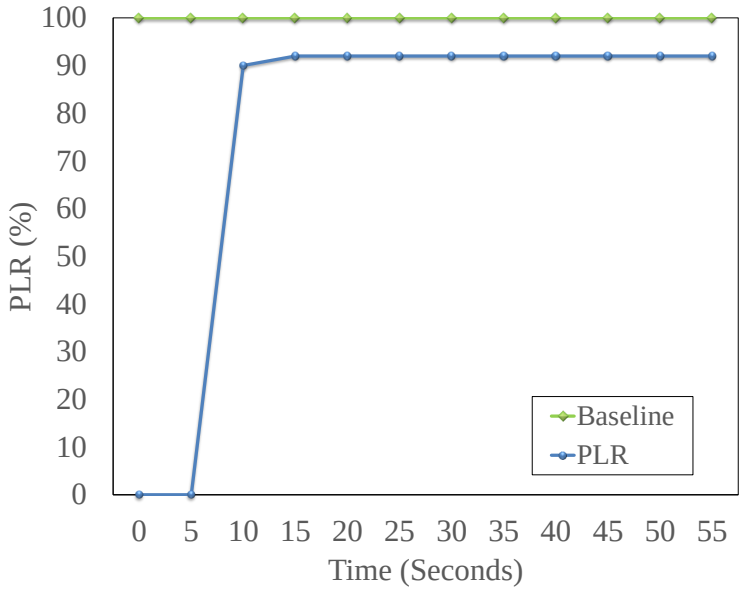

Figure 5.14: Percentage of positive likelihood ratio under time analysis attacks

than ours in unpopular case; however, this case is not preferred by attackers because they must choose and generate requests for just unpopular contents.

For an attacker, sending requests for unavailable contents is much easier than for available contents. In routing attack experiments, an attacker can achieve similar impact when the attacker sends malicious requests for available or unavailable requests. The reason behind this is that ICN tries to satisfy requests from different paths that prevents legitimate requests from being satisfied. This scenario is similar to sending malicious requests for available contents preventing legitimate contents from being delivered.

In caching experiments, cache hit ratio is enhanced over the baseline as we apply our ranking technique, detection and prevention phases. ICN caching now caches the most popular contents and evicts the least popular ones. In normal ICN behavior, each node caches any content passing through this node in on-path caching techniques. In off-path caching techniques, neighbouring nodes also can cache these contents. Using our solution,

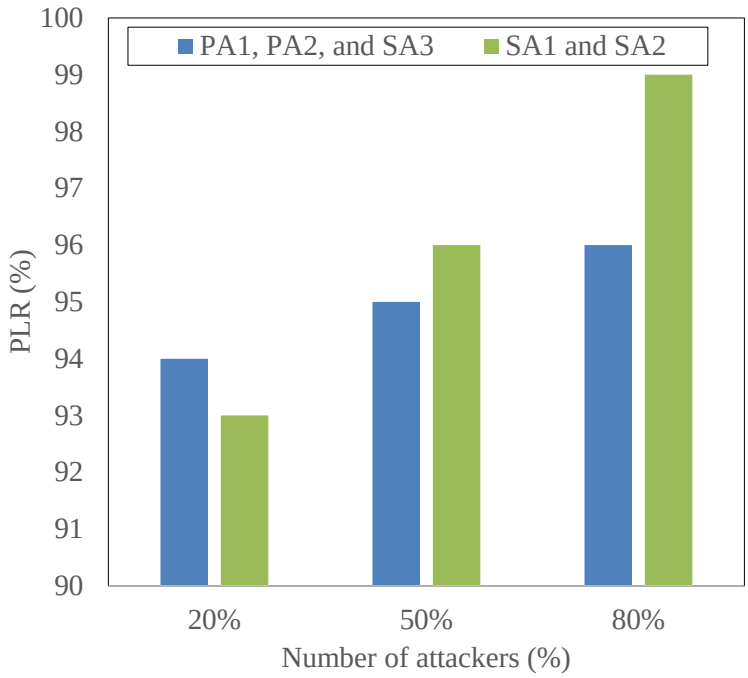

Figure 5.15: Percentage of positive likelihood ratio under routing attacks

we decrease caching rate for unpopular contents and in case of any cache replacements, the least popular ones will be evicted. This is the main reason for our enhancement over the baseline case. As depicted in Figure 5.12, unpopular distribution is the simplest caching attack to be detected. Uniform distribution is considered to be the most difficult caching attack to be detected. The primary reason is that cache hit ratio of an attacker in the uniform distribution is greater than the cache hit ratio in the unpopular distribution.

The difference between our results and baseline values, as shown in Figure 5.15 and Figure 5.16, is due to our threshold values and user behaviors before applying our counter-measures. This difference can be decreased, if threshold values are chosen to include all or most of the legitimate users. While an increase of the threshold values can decrease false positives, it may also increase false negatives at the same time.

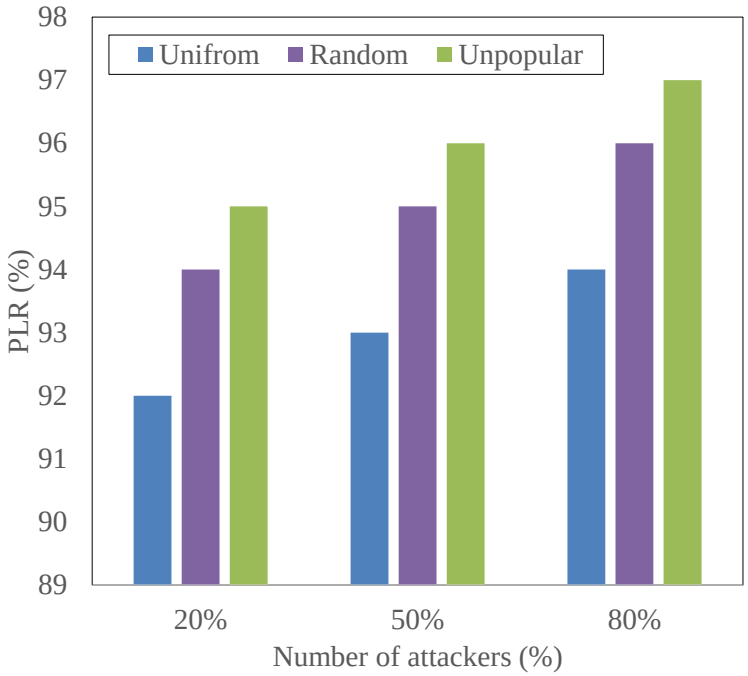

Figure 5.16: Percentage of positive likelihood ratio under caching attacks

5.6 Summary

In this chapter, we present different scenarios of DDoS attacks in ICN routing and caching. We also present our solution (DADI) that detects and prevents these DDoS attack scenarios. We start by calculating threshold values for Request Satisfaction Ratio (RSR), request rate, cache hit ratio, rating for contents and rating for publishers. We compare user request rate and pattern with respect to these threshold values in the detection phase. Then we prevent these attacks using appropriate actions based on the attack case. In caching related DDoS attacks, we replace cached content based on rating for cached contents and least recently used contents. Our results show that the solution succeeds in mitigating these attacks and enhances ICN performance in the existence of all attacks.

Chapter 6

Preventing Unauthorized Access in ICN

6.1 Introduction

ICN has in-network caching capability, which enables any node to cache any content coming from any publisher. ICN subscribers can access contents from different distributed locations. This in-network caching capability maximizes the problem of unauthorized access to ICN contents. Existing access control mechanisms cannot be applied directly to ICN architectures because of the following three reasons: ICN supports the in-network caching, ICN requests do not have any user identification information, and ICN does not depend on IP addresses. Also, existing access control mechanisms fail to address the following attributes of a good access control mechanism: 1) reduce unnecessary communication overhead, 2) eliminate modifications to ICN architectures, 3) minimize the exchange of secret keys, 4) do not require large number of extra operations for access control, 5) prevent access control attacks, 6) preserve the privacy of ICN users, 7) ensure the integrity of the retrieved content, and 8) can be applied with different ICN naming schemes (self-certifying and hierarchical) [6, 11, 35, 38, 45, 54, 62, 70, 72, 80, 82].

In this chapter, we are concerned with unauthorized access attacks. Access control

mechanisms can be classified as centralized, decentralized, and encryption-based mechanisms. In centralized access control, there are extra entities such as authentication servers or key generation and distribution centers. These entities are responsible for evaluating ICN users against access control policies for ICN contents. In decentralized access control mechanisms, ICN subscribers and nodes work together for mutually authenticating each other and ensure that legitimate users access legitimate contents. In the latter case, ICN publishers are also included in this authentication process. In the encryption-based mechanisms, access control is satisfied by encrypting ICN contents or requests or both.

In order to prevent unauthorized access attacks, we first propose a Decentralized Access Control Protocol for ICN (DACPI) that uses RSA and Diffie-Hellman techniques. Then we propose a decentralized **E**lliptic **C**urve based **A**ccess **C**ontrol **(ECAC)**. To the best of our knowledge, this chapter presents the first use of elliptic curve cryptography for access control in ICN architectures. Elliptic Curve (EC) can be used to achieve encryption and decryption goal and key exchange goal. Elliptic curve offers equal security for a smaller key size compared to the well-known RSA technique, hence reducing processing overhead. For example, key size of 112 bits in EC is equivalent to key size of 512 bits in RSA; key size of 512 bits in EC is equivalent to key size of 15,360 bits in RSA. Also, ECC is considered in standardization efforts such as the IEEE P1363 standard for public key cryptography [73]. ECAC uses the following techniques: elliptic curve, hashing, and random number generations. The main objective of ECAC protocol is to allow only legitimate users to access legitimate contents.

Our goal is to prevent unauthorized access attacks in ICN, achieving the following explicit design goals:

- Access control: We design ECAC to prevent attackers from accessing ICN contents

and only allow legitimate users access ICN restricted access contents.

- Minimal change to ICN architectures: Our proposed protocol does not require any extra entities or changes to ICN architectures. We only add authentication messages and cryptographic operations for access control.

- ICN utilization: Any access control protocol requires extra overhead to the architecture. We propose a protocol that minimizes the required overhead. We use different metrics such as number of public messages and request access time delay to measure the required overhead based on our solutions.

Our contributions in this chapter can be summarized as follows:

- We propose a decentralized access control protocol (DACPI) depending on RSA and Diffie-Hellman cryptographic techniques and then we propose an enhanced version that achieve better performance results based on elliptic curve cryptography (ECAC).

- We prevent unauthorized access attacks in ICN using fewer public messages with respect to the existing solutions.

- We show the effectiveness of ECAC by performing security and performance analysis and comparing ECAC with the related protocols.

We evaluate ECAC using various scenarios and under different request rates and number of attackers with respect to the number of legitimate users. We measure ICN performance metrics in the presence of and without ECAC, a decentralized access control mechanism (DACPI), and a representative of centralized access control mechanism (ACED) [35]. Our results show that the ECAC requires less overhead than the Decentralized Access Control Protocol for ICN (DACPI) in terms of ICN request access time delay and

almost achieves results similar to the centralized Access Control Enforcement Delegation (ACED). However, ACED requires more public access control messages and architecture modifications

The remainder of this chapter is organized as follows. Section 6.2 presents the related work in ICN access control literature and shows generic access control mechanisms. Section 6.3 presents our first proposed decentralized access control protocol (DACPI). Section 6.4 presents the proposed decentralized elliptic curve based access control protocol (ECAC). This section shows ECAC specifications, internal steps, and ECAC analysis. In Section 6.5, we analyse ECAC attributes with respect to the good ones. In Section 6.6, we compare ECAC protocol with existing access control protocols. Section 6.7 presents our experiments and compares between ECAC and existing access control mechanisms. Finally Section 6.8 presents our conclusions.

6.2 Related Work and Background

6.2.1 Related Work

We discuss the existing specific access control mechanisms in ICN in the context of our proposed protocol. The existing work controls the access to ICN contents by different ways such as centralized [11, 35], decentralized [6, 80], and encryption-based [38, 45, 54, 62, 72, 82].

For the centralized access control techniques, Fotiou et al. [35] propose an access control mechanism (ACED) that evaluates subscriber requests against access control policies. This mechanism uses extra entities such as a relaying party and an access control provider. Relaying party is responsible for content distribution to legitimate users. Access control provider allows publishers to create contents with access control policy and

evaluates subscriber requests. Aiash et al. [11] present an authentication and authorization mechanism for NetInf ICN architecture. This solution uses an extra entity named trusted ticket granting, which is responsible for key generation and distribution. This solution uses an ID-based cryptographic technique for securing the messages. The name resolution system in NetInf is responsible for authentication for ICN users and contents publication and subscription.

For the decentralized access control mechanisms, Wang et al. [80] propose a session-based authentication mechanism that can be applied to different ICN architectures. In this mechanism [80], the authors propose two names for each ICN content, one is public and the other is a secure name that is known only by legitimate users. The mechanism depends on symmetric keys that need to be generated and distributed in a secure manner. In our earlier work, we proposed DACPI [6], which is a decentralized access control protocol for ICN. DACPI is based on ICN self-certifying naming scheme. ICN self-certifying naming is unique, persistent, not limited to any organization and makes it easier for integrity checking. In this protocol, we use the following cryptographic techniques: RSA public key infrastructure, key exchange using Diffie-Hellman, hashing, and random number generations. Some of the public messages are encrypted using public key of the receiver and then using the private key of the sender to achieve authenticity and confidentiality. DACPI succeeds to decrease the number of public messages required for access control to five messages. The related access control protocols either centralized or decentralized require more than five messages. DACPI does not require any extra entities to ICN architectures. DACPI is designed to satisfy the attributes of good access control mechanisms.

For the encryption-based solutions, Kurihara et al. [54] present an access control framework for content centric networking (CCN-AC). This framework depends on symmetric,

asymmetric, and nonce keys to encrypt and decrypt the manifest and content objects. The framework uses extra entities such as a key manager and access policy manager and require modifications to the content metadata. Ghali et al. [38] describe an interest-based access control for content centric networking. This solution hides content names from unauthorized subscribers by using encryption-based name obfuscation. Misra et al. [62] use a broadcast encryption technique for a framework to deliver contents to legitimate users. ICN contents are encrypted with symmetric keys and these keys are distributed to ICN users by session keys. Ion et al. [45] propose attribute-based encryption to achieve access control and can also be used in ICN routing. Wood et al. [82] propose an encryption-based technique to personalize ICN cached contents to each ICN legitimate user. Smetters et al. [72] present an access control mechanism for the previous version of named data networking architecture, which depends on a group-based access control.

There are also other work related to access control in ICN. Rembarz et al. [70] separate the information objects in private domain to prevent the unauthorized access. In [81, 92], the authors propose a secure naming scheme that can be used for access control services. Malicious access privacy issues based on timing analysis attacks and related countermeasures are handled in [63]. In this paper, an attacker violates subscriber's privacy by measuring the time differences between cached and uncached contents, which can be used as an indicator, if the subscriber has requested this content before or not. Burke et al. [18] provide an example in building autonomous systems using NDN architecture by authenticating the lighting control.

Our proposed access control protocol (ECAC) falls in the decentralized access control mechanism category. ECAC does not require extra entities or architecture modifications like the centralized mechanisms or content modifications similar to the encryption-based

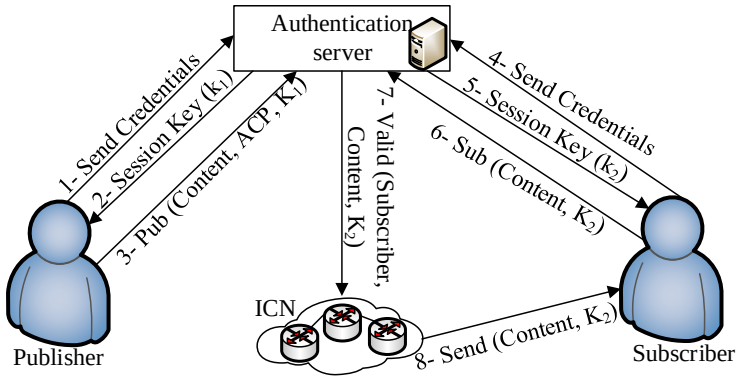

Figure 6.1: Generic centralized access control scheme in ICN

techniques. ECAC generates superior performance results than those provided by our previous work (DACPI). Generally, elliptic curve cryptography has been used in different technologies such as RFID, wireless sensor networks, vehicular networks, and multimedia encryption [14, 22, 33, 43, 56, 75, 76]. New standards are recommending elliptic curve based cryptography such as zigbee networking, intelligent transportation systems (ITS), and Suite B Cryptography Standards [59].

6.2.2 Centralized Access Control Mechanisms

In the ICN centralized access control mechanisms [11, 35], there are extra entities responsible for achieving access control services. All ICN requests go through a central authority that accepts or rejects the requests. This mechanism simplifies the access control, however, no access can be granted if the central authority fails. There are some variations to generic centralized mechanisms. One mechanism may use more than one type of extra servers. Some of these extra servers are authentication servers that are responsible for managing access control policies and the other servers are responsible for managing ICN nodes to send ICN contents to legitimate users. Another mechanism may use key distribution center

to distribute secret keys for ICN users and depend on ICN name resolution systems for managing contents and access control policies.

Figure 6.1 shows a generic scheme for a centralized access control mechanism in ICN. In this figure, the authentication server indicates the extra entity responsible for access control policy or key distribution.

First, a publisher sends his credentials to the authentication server and the authentication server replies with a session key, if the publisher is a legitimate user. Then the publisher sends a publication message with the ICN content name, access control policy, and the session key. A subscriber also starts by sending his credentials and the authentication server replies with another session key, if the person is a legitimate user. Then the subscriber sends a subscription message using the ICN content name and the session key. The authentication server evaluates the subscriber's request against the access control policy attached with the ICN content. If the subscriber is an authenticated user for this content, the authentication server sends a message to the ICN to forward the content from the best available copy to this subscriber. In this generic scheme, the secret keys (k_1, k_2) can also be used in different ways to encrypt and decrypt ICN contents and requests and the keys are distributed in a secure way.

Centralized access control mechanisms prevent unauthorized access without adding more overhead to ICN contents, which enhances ICN performance in the existences of these mechanisms. The delay, cost, and administration overheads are coming from the communications between the required new entities and architectural modifications for access control.

6.2.3 Decentralized Access Control Mechanisms

In the ICN decentralized access control mechanism [6, 80], there is no need for extra entities or architecture modifications. ICN users and nodes collaborate for mutual authentication. The access control responsibilities are distributed across many locations. The decision of accepting or rejecting an ICN request can be taken by the ICN nodes themselves.

This approach is more stable because no single point of failure exists. If one ICN node fails, the request can be redirected to any other node. There are also some variations that can be used for access control. Many security techniques can be used such as public key infrastructure, identity-based cryptography, hashing, shared secrets generation and distribution, to name a few.

Figure 6.2 depicts a generic decentralized access control scheme. A publisher sends a message with the content name, hashing value, secret information, and public parameters. Hashing value is used for integrity checking, while secret and public parameters are used for confidentiality. A subscriber sends an encrypted message with the content name that achieves both authenticity and confidentiality, and ICN nodes forward the request to the responsible publisher. Then the publisher replies with an encrypted message that also achieves both authenticity and confidentiality with a nonce and public parameters to the subscriber. The subscriber afterwards sends a request with the content name, hashing value, and public parameters to ICN nodes. ICN nodes evaluate the request with respect to the received publication and then sends the content back to the subscriber. Finally the subscriber also evaluates the content and secret information to be sure that everything is authenticated.

Unauthorized access attacks can be prevented using decentralized access control mechanisms without the need to modify ICN architectures. In these mechanisms, the delay

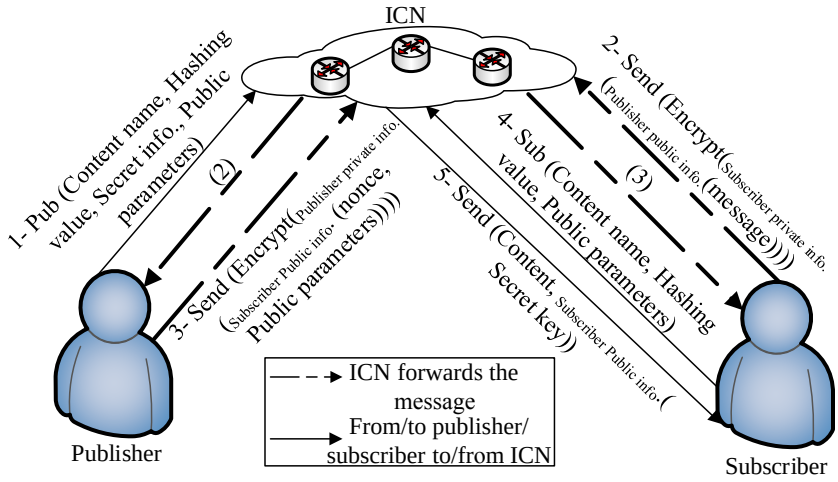

Figure 6.2: Generic decentralized access control scheme in ICN

is coming from the used security techniques for access control purposes. If the security techniques are used in an optimized way, the decentralized mechanisms can achieve performance results close to centralized ones without architectural modifications. In our point of view, this advantage makes decentralized access control mechanisms better than centralized ones.

6.3 Decentralized Access Control Protocol for ICN (DACPI)

In this section, we present the proposed protocol (DACPI) in detail. We start with the ICN reference model that we have used in our protocol. In subsection 6.3.1, we present DACPI specifications. In subsection 6.3.2, we show DACPI internal steps that are carried out by subscribers and ICN nodes. In the following subsections, we analyse DACPI.

To support DACPI, ICN routers should perform the following extra tasks: compare between two hashing values from a subscriber and a publisher; calculate a shared key. Additionally, the metadata associated with the content contains more details than the ICN

Table 6.1: DACPI notations

Notation	Definition
n_1, n_2	Nonces generated by a publisher and subscriber, respectively
p	Public large prime number used as the base for key exchange
a	Primitve root used for key exchange
PU_{pub}, PR_{pub}	Public and private key pairs for a publisher
PU_{sub}, PR_{sub}	Public and private key pairs for a subscriber
x	Secret number selected by a publisher
y	Secret number selected by a subscriber
X	Public number calculated by a publisher
Y	Public number calculated by a subscriber
S	Shared key

normal metadata. Our protocol's metadata contains hashing value, nonces and other secret and public parameters.

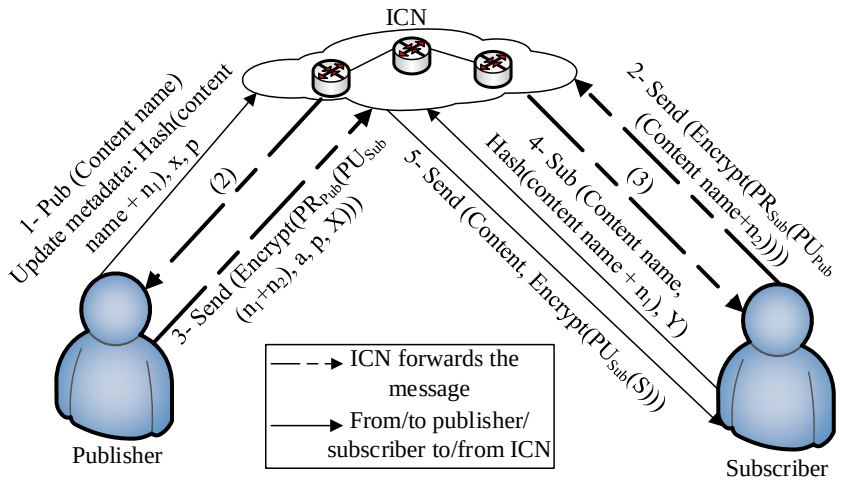

Figure 6.3: Proposed decentralized access control protocol in ICN (DACPI)

6.3.1 DACPI Specifications

In this subsection, we describe the proposed decentralized access control protocol (DACPI). The protocol depends on RSA public key infrastructure and uses Diffie-Hellman for key exchange. Some of the public messages are encrypted using public key of the receiver and then using the private key of the sender to achieve authenticity and confidentiality. In DACPI, we depend on self-certifying naming scheme and use its security features. In ICN self-certifying naming scheme, names consist of two parts of the form *P:L* and metadata associated with each content. The two parts represent a unique ICN content name. The metadata is used for authenticity and integrity checking. As mentioned in Chapter 2, The first part (*P*) is the cryptographic hash of the owner's public key. The second part (*L*) is a content label assigned by the owner. Metadata contains the full public key and digital digest signed by the owner. A subscriber requests the content with both parts *P* and *L* and receives the content, publisher's public key and signed hash value. The subscriber then checks that the hash valve of the received public key is the same as the sent hash value to ensure that the content is received from an authenticated publisher. The subscriber also decrypts the received signed hash and compares it with the hash of the content to ensure that the integrity is satisfied. Table 6.1 shows the notations used in the proposed protocol.

In DACPI, as shown in step 1 of Figure 6.3, the publisher sends a publication message with the content name as in the self-certifying naming scheme that consists of *P:L*. The publisher updates the metadata attached with the content with the following: hashing of content and nonce (n_1), parameters *(x)* and *(p)*. The addition in the hashing function means concatenation. The content name is the only published part. The remaining parameters in the metadata are used in later steps for access control purposes.

Pub (Content name)

Update metadata: Hash(content name + n_1), x, p

Then in step 2, the subscriber sends a message with the content name and nonce (n_2) encrypted using publisher's public key and subscriber's private key. ICN forwards the message to the publisher.

Send (Encrypt(PR_{Sub}(PU_{Pub}(content name + n_2))))

The publisher decrypts the message using subscriber's public key and publisher's private key and extracts nonce (n_2). In step 3, The publisher replies with an encrypted message with the subscriber's public key and then publisher's private key. The reply message contains two nonces (n_1) and (n_2), primitive root (a), public parameters (p) and (X). ICN forwards the reply message to the subscriber.

Send (Encrypt(PR_{Pub}(PU_{Sub} (n_1 + n_2), a, p, X)))

The subscriber decrypts the message using publisher's public key and using subscriber's private key and extracts nonce (n_1). The subscriber also calculates public parameter (Y) and key (S).

Calculate Hash(content name + n_1)

$Y = a^y \bmod p$

$S = X^y \bmod p$

In step 4, the subscriber sends a subscription message to ICN with the content name, hash value of content and nonce (n_1), and public parameter (Y).

Sub (Content name, Hash(content name + n_1), Y)

ICN then evaluates the subscription message and calculates key (S).

Compare two hashes received from publisher and subscriber

$S = Y^x \bmod p$

Then in step 5, ICN sends the content back to the subscriber with encrypted key (S), if

the subscriber is an authenticated user.

Send (Content, Encrypt(PU_{Sub}(S)))

The subscriber evaluates the reply from ICN and accepts the content, if it comes from an authenticated node.

6.3.2 DACPI Internal Steps

In DACPI, there are extra internal steps that are carried out by ICN subscribers and nodes. These extra steps are used for evaluating the messages between subscribers and nodes. Figure 6.4 shows the sequence diagram of DACPI between ICN publishers, subscribers, and nodes. In this figure, we focus on the extra internal steps, other steps are explained in detail in Figure 6.3. In step *i1*, the subscriber calculates the hash value of the content and nonce (n_1), and the public parameter (Y) to send these values in the subsequent public message. The subscriber also calculates the shared key (S_{sub}). In step *i2*, the ICN node compares the two hashes and also calculates the shared key (S_{ICN}). If the two hashes are the same, then ICN node sends the content to this legitimate subscriber. In step *i3*, the subscriber compares between the received and calculated key. If the two keys are the same and self-certifying naming verification is valid, then the subscriber ensures that the content is coming from an authenticated node.

6.3.3 Security Analysis

In the proposed protocol, using only the public parameters transmitted in the public messages is not sufficient for an attacker to calculate the secret values. In this subsection, we provide a comprehensive list of attacks that may happen in ICN access control schemes. For DACPI, we explore the following: man-in-the-middle, forward security, and replay

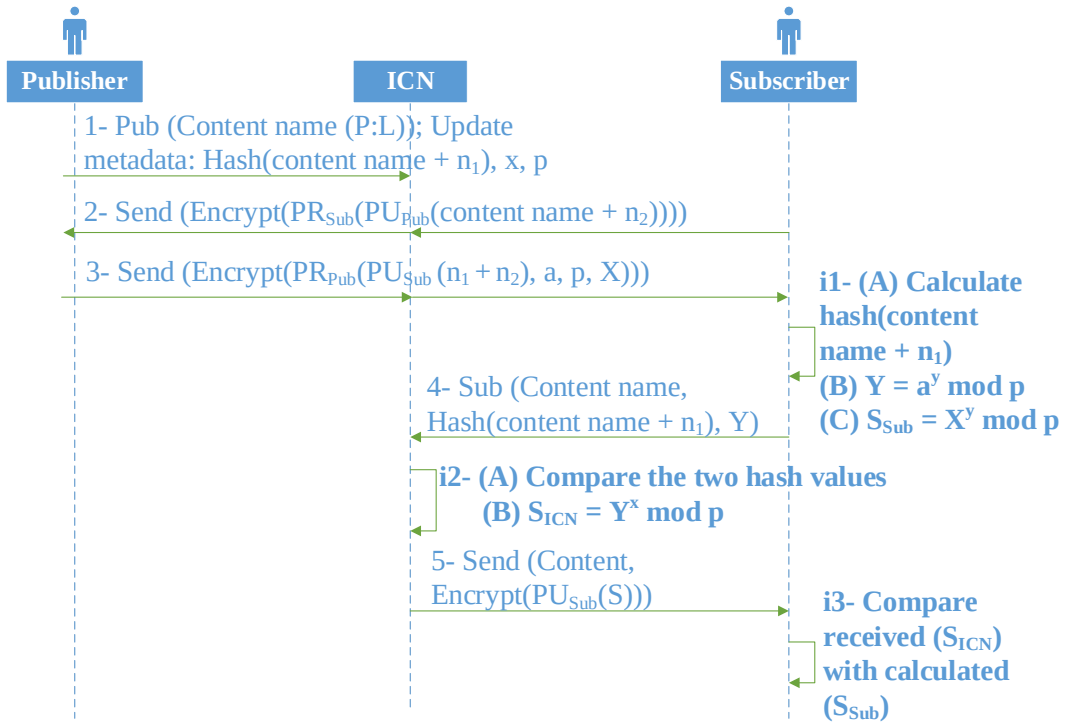

Figure 6.4: Sequence diagram of DACPI

attacks. Additionally, for ICN architectures, we explore the following: ICN content or request modifications and privacy violations for ICN users.

Man-in-the-middle attacks. An attacker plays the classical man-in-the-middle attack to impersonate an authenticated subscriber or publisher or ICN node. To impersonate the subscriber, an attacker needs to forward the subscription messages to persuade ICN publisher and node that he/she is an authenticated subscriber. These messages will be invalid, because they must be encrypted using subscriber's private key and different nonce (n_2) is used with every subscription request. Additionally, an attacker also can impersonate an authenticated publisher or ICN node to persuade ICN subscriber that he/she is an authenticated publisher or ICN node. This will also be invalid, as the attacker must know the publisher's private

key and published nonce (n_1) with the ICN content and initial information (x) to correctly communicate with the subscriber. The secret keys and nonces cannot be easily detected by the man-in-the-middle attacker.

Forward security. This is a feature that ensures that past communications between ICN nodes and subscribers are secure, even if one communication is compromised. An attacker cannot conclude the previous requests, even if the attacker succeeds to attack one communication. Forward security is guaranteed in this protocol, as our mutual authentication depends on using private keys of publishers (PR_{pub}) and subscribers (PR_{sub}), shared key (S), and nonces (n_1, n_2). Private and shared keys depend on large prime numbers that cannot be easily broken. New nonces are used for each content publication and subscription request.

Replay attacks. An attacker can store the public messages used in the protocol and replay these messages in a later stage to access ICN contents. The attacker will not be able to form valid messages as the protocol uses different nonce with each request, and hence the attack is unsuccessful.

ICN content or request modifications. The access control protocol depends on self-certifying naming scheme, which ensures data integrity for ICN contents. An attacker can try to modify the public messages during transmission. This attack cannot be done under our protocol for two reasons. First, only an authenticated subscriber, who has a private key (PR_{sub}), nonce (n_1), and shared key (s) can form a valid message. Second, only authenticated node, which has information (x), nonce (n_1) can form a valid message. Both ICN node and subscriber check the data integrity before sending and receiving the ICN contents.

Privacy violations of ICN users. An attacker monitors the public messages to gain private information about content popularities and who requested which content. All parameters

in the public messages introduced by our protocol are: encrypted using RSA public key infrastructure; converted by one-way hashing value; public parameters that cannot be used to determine the secret values.

6.3.4 Performance Analysis

Communication overhead. In DACPI, we achieve mutual authentication between the ICN node and subscriber in five public messages. Additionally, there are two more steps done on the subscriber side and one step on the ICN node side. These additional steps are not considered as communication overhead since they are not transmitted. The size of the communication messages is increased compared to those in the original ICN architectures, because we have added some encrypted information, nonces, and secret keys.

Computational overhead. In DACPI, we build our access control techniques based on RSA public key infrastructure, shared key based on Diffie-Hellman key exchange, random number generation and hashing techniques. These techniques represent the additional computational overhead over the traditional ICN architectures. Although there is a relatively high computational overhead, the security level of the access control protocol in ICN architecture and public messages are remarkably increased.

Storage requirements. In DACPI, no centralized storage is required. However, an ICN publisher should maintain their users with access rights. This point is practically valid, as currently Internet users need to register for different content owners. Based on the registration step, each owner builds a profile for a user with privileges that can be used by the user.

6.4 Elliptic Curve based Access Control Protocol (ECAC)

In this section, we present our recommended proposed protocol (ECAC) based on elliptic curve cryptography. We start with ECAC specifications, then we present the internal steps of the proposed protocol. We also perform security analysis for ECAC for the same attacks. In the last subsection, we do a performance analysis for ECAC.

6.4.1 ECAC Specifications

In this subsection, we describe the proposed decentralized access control protocol (ECAC). ECAC uses elliptic curve cryptography for two purposes: encryption and decryption, and key exchange. Elliptic curve cryptography (ECC) is an efficient cryptographic technique for public key infrastructures (PKI). The security of ECC comes from the elliptic curve logarithm problem, which means the difficulty of calculating discrete logarithms in a group of points defined over a finite field on an elliptic curve. Public messages in ECAC are encrypted using the public key of the receiver and a random positive integer chosen by the sender. In ECAC, we depend on the self-certifying naming scheme. Table 6.2 shows the notations used in the ECAC. In ECAC, an ICN publisher uses elliptic curve encryption for plaintext $PubP_m$ to generate cipher text $PubC_m$ by the following equation:

$$PubC_m = (k_1G,\ PubP_m + k_1P_{sub})$$

Similarly, an ICN subscriber uses elliptic curve encryption for plaintext $SubP_m$ to generate cipher text $SubC_m$ using the following equation:

$$SubC_m = (k_2G,\ SubP_m + k_2P_{pub})$$

The ICN publisher receives subscriber's secure message and decrypts the message to retrieve subscriber's plaintext by the following equations:

$$SubP_m = SubP_m + k_2P_{pub} - n_{pub} * (k_2G)$$

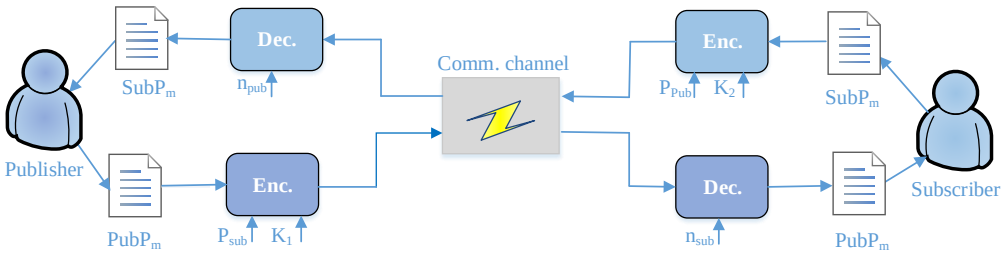

Figure 6.5: Encryption and decryption processes in ECAC: A subscriber encrypts plaintext $SubP_m$ using publisher's public key P_{pub} and secret number k_2 and the publisher decrypts the message using the publisher's private key n_{pub}. A publisher encrypts plaintext $PubP_m$ using subscriber's public key P_{sub} and secret number k_1 and the subscriber decrypts the message using the subscriber's private key n_{sub}. The two keys k_1 and k_2 are not transmitted between ICN publishers and subscribers

$$SubP_m = SubP_m + k_2 * (n_{pub}G) - n_{pub} * (k_2G)$$

Similarly, the ICN subscriber receives publisher's secure message and decrypts the message to retrieve publisher's plaintext by the following equations:

$$PubP_m = PubP_m + k_1P_{sub} - n_{sub} * (k_1G)$$

$$PubP_m = PubP_m + k_1 * (n_{sub}G) - n_{sub} * (k_1G)$$

For both publisher and subscriber, elliptic curve shared key calculations can be calculated as follows:

$$S = x * Y = x * (y * G) = y * (x * G) = y * X$$

Figure 6.5 shows the encryption and decryption processes in ECAC. In the following paragraphs, as shown in Figure 6.6, we describe ECAC's steps in detail.

Step 1: *Pub (Content name)*

Update metadata: Hash(content name + n_1), x, q, a, b, G

ICN publisher sends a publication message with the content name consisting of the *P:L* parts as in self-certifying naming. The metadata attached with the content is updated with

Table 6.2: ECAC notations

Notation	Definition
n_1, n_2	Nonces generated by a publisher and subscriber, respectively
q, a, b, G	Elliptic curve public parameters
P_{pub}, n_{pub}	Public and private key pairs for a publisher
$PubP_m$, $PubC_m$	Plaintext and cipher text of a publisher
P_{sub}, n_{sub}	Public and private key pairs for a subscriber
$SubP_m$, $SubC_m$	Plaintext and cipher text of a subscriber
k_1, k_2	Random numbers generated by a publisher and subscriber, respectively
x, X	Publisher's secret and public keys for key exchange
y, Y	Subscriber's secret and public keys for key exchange
S	Shared key

the following information: hashing value of content and nonce (n_1), parameter (x) and elliptic curve public parameters (q, a, b, G), where q is a prime or an integer of the form 2^m and G is a point on elliptic curve whose order is large value n such that $nG = 0$. The addition in the hashing function means concatenation. In ECAC, elliptic curve encryption and decryption achieve messages confidentiality, while shared key, hashing technique, self-certifying naming achieves content authenticity and integrity.

Step 2: *Send (Encrypt(P_{Pub},k_2(content name + n_2)))*

The subscriber encrypts a message using publisher's public key and random positive integer k_2. The value of k_2 will not be known to anybody except the subscriber. The message itself contains the content name and nonce (n_2). ICN routers forward the message to the publisher.

Step 3: *Send (Encrypt(P_{Sub},k_1 (n_1 + n_2), q, a, b, G, X))*

In order to extract nonce (n_2), the publisher decrypts the message using publisher's

private key without knowing the value of k_2. Then the publisher sends another message encrypted using subscriber's public key and random positive integer k_1. The value of k_1 will not be known to anybody except the publisher. The message itself contains two nonces (n_1, n_2), elliptic curve public parameters, and publisher's public key used for key exchange. ICN routers forward the message to the subscriber.

In order to extract nonce (n_1), the subscriber decrypts the message using subscriber's private key without knowing the value of k_1. The subscriber also calculates public parameter (Y) and key (S), as shown in the following internal steps:

Calculate hash(content name + n_1)

$Y = y * G$

$S = y * X$

Step 4: *Sub (Content name, Hash(content name + n_1), Y)*

The subscriber sends a subscription message to ICN edge router with the content name, hash value of content and nonce (n_1), and public parameter (Y). In order to validate subscriber's request, ICN edge router then evaluates the message and calculates key (S), as shown in the following internal steps:

Compare two hash values from publisher and subscriber

$S = x * Y$

Step 5: *Send (Content, Encrypt(P_{Sub}(S)))*

If the subscriber is verified as an authenticated user, then the ICN edge router sends the requested content to the subscriber. This content is sent with the shared key (S) encrypted using subscriber's public key. Finally, the subscriber evaluates the reply from ICN edge router and accepts the content, if it comes from an authenticated node.

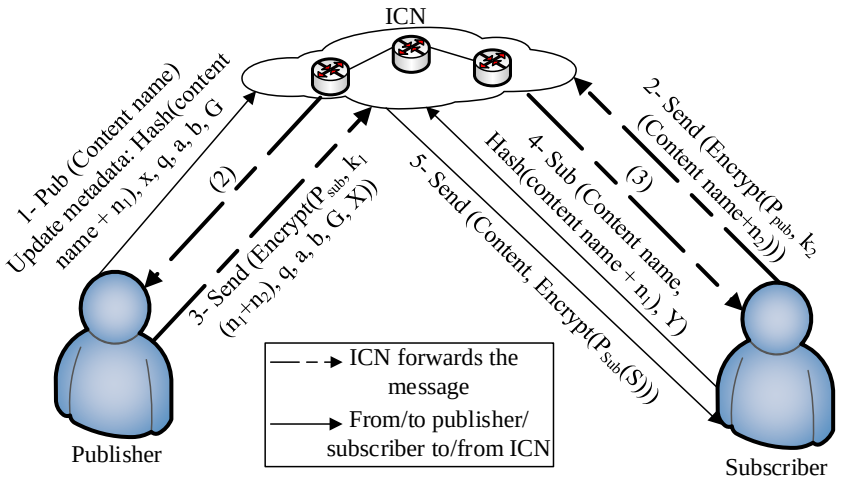

Figure 6.6: Proposed decentralized access control protocol in ICN (ECAC)

6.4.2 ECAC Internal Steps

We describe the required internal steps in ECAC in order to achieve access control for restricted-access contents in ICN architectures. The purpose of these steps is to evaluate the public messages exchanged between subscribers and ICN routers and do some calculations. The sequence diagram of ECAC between ICN publishers, subscribers, and nodes is depicted in Figure 6.7. The goal of the first internal step *i1* is to enable the subscriber to extract nonce n_1 to formulate the hash value of content name plus the extracted nonce. Then the subscriber calculates public number Y for key exchange purpose that will be sent later in the protocol. This public parameter is calculated using subscriber's private key that is used for key exchange and public parameter G. The subscriber afterwards calculates the shared key *(S)*. The aim of the second internal step *i2* is to evaluate the subscriber's request to check whether the subscriber is an authenticated user or not. ICN edge router compares between its hash value and the received hash value from the subscriber and then

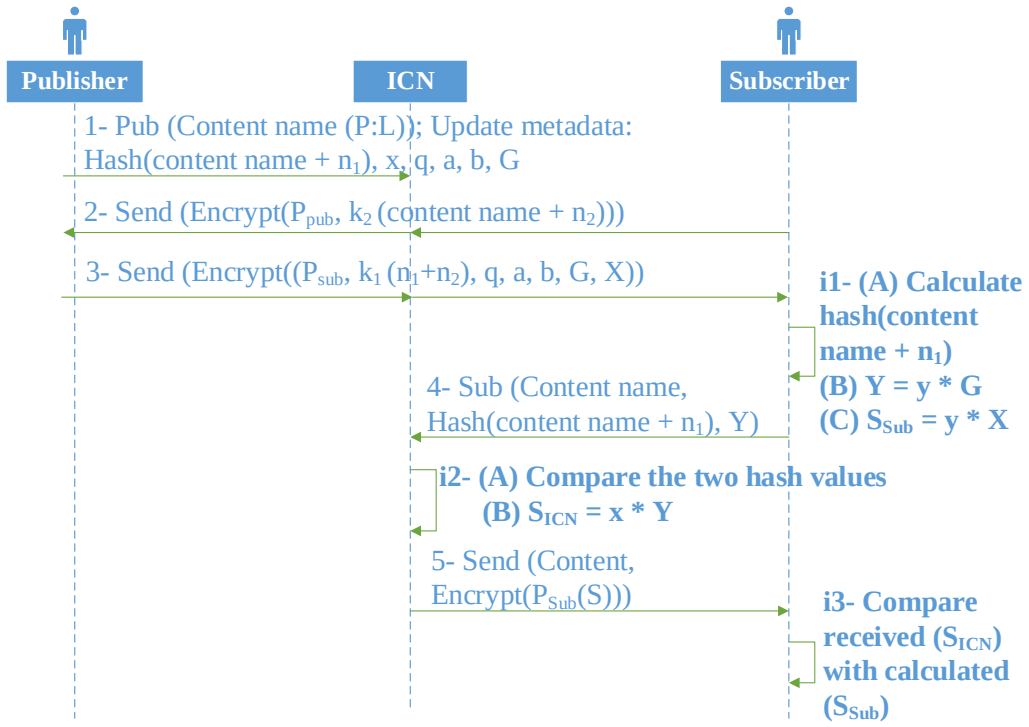

Figure 6.7: Sequence diagram of ECAC

calculates the shared key *(S)*. In the third internal step *i3*, the subscriber evaluates the response from ICN edge router to check its validity and authenticity by comparing between the received shared key with subscriber's calculated key in addition to ICN self-certifying name checking.

6.4.3 Security Analysis

Using only the public parameters transmitted in the public messages in ECAC is not enough for an attacker to gain access to the restricted-access ICN contents. In this subsection, we explore the same comprehensive list of attacks that may happen in ICN access control mechanisms.

Man-in-the-middle attacks. To impersonate an ICN subscriber, an attacker needs to know subscriber's private keys and elliptic curve random number (k_2). To impersonate ICN publisher or an edge router, an attacker needs to know publisher's private keys and elliptic curve random number (k_1). Additionally, the attacker needs to know the used nonces (n_1, n_2), which are different with each content request and shared keys. Random numbers in elliptic curve cryptography even are not transmitted between different users. The attacker needs this secret information in order to form valid messages to perform unauthorized access attacks. Using the public parameters used in ECAC, the attacker cannot get this secret information. Consequently, the attacker's messages will be invalid and the attacks will be unsuccessful and easily detectable. The difficulty of determining secret number (k_1, k_2) comes from the discrete logarithm problem in EC. In EC, the addition operation associates to each pair of points on EC to obey the abelian group idioms. Multiplication is defined as repeated addition, where the addition is performed over an EC. The addition between two points on the elliptic curve depends on the selection of the q parameter.

$a * k = a + a + a$ ... *(k times)*

Cryptanalysis involves determining k given a and $a*k$, which is the discrete logarithm problem in EC.

Forward security. Using ECAC, past communications are secure, even if an attacker succeeds to get unauthorized access to one ICN content because these past communications use different nonces (n_1, n_2) and different random numbers (k_1, k_2). Also, ECAC uses elliptic curve private keys that depends on large prime numbers or an integer of the form 2^m. These keys in ECAC cannot be easily calculated using the available public information in the protocol.

Replay attacks. Traditionally, an attacker stores public messages and replays them again

in a later stage to persuade the other party that it is an authenticated entity. This type of attack is prevented in ECAC due to the continuous change of the used parameters with each message.

ICN content or request modifications. If an ICN subscriber or an ICN edge router finds any change in ICN content or public messages, then the communication is stopped and one party detects that the other is not a legitimate entity. For content integrity verification, ECAC depends on ICN self-certifying scheme to do this. For messages verifications, both parties cannot form a valid message without knowing the required secret parameters. Using ECAC, content and request integrity is verified and any modifications can be easily detected.

Privacy violations of ICN users. The main target of an attacker in this type of attacks, is to calculate used secret keys in order to get access to private information. This private information enables the attacker to know about content popularities and requested ICN users. Using ECAC, an attacker cannot determine these secrets due to all aforementioned reasons including elliptic curve cryptography, nonces, random numbers and secret keys.

6.4.4 Performance Analysis

In this subsection, we analyse communication, computational, and storage overhead caused by ECAC on ICN architecture. In Section 6.7, we experimentally evaluate how these overheads affect ICN performance.

Communication overhead. In ECAC, we achieve mutual authentication between ICN edge routers and subscribers in five public messages as in DACPI. Other centralized and decentralized mechanisms require more authentication messages than our proposed protocol. In order to calculate shared key and encrypt/decrypt public messages, there are

three internal steps that are not considered as communication overhead because they are not transmitted between ICN routers and subscribers. We include more parameters and encrypted information to ICN public messages, which lead to an increased size of these access control communication messages.

Computational overhead. In ECAC, we build our access control technique based on elliptic curve cryptography which is used for key exchange and encryption/decryption, random number generation and hashing techniques. The security level of access control in ICN is increased, using these extra techniques.

Storage requirements. Although there is no need for extra entities in ECAC, there is extra required storage in ICN publishers and routers to maintain the additional parameters. For ICN publisher, extra storage is required for storing elliptic curve public parameters and nonces. For ICN edge router, extra storage is needed to store elliptic curve public parameters and shared key. Using this light storage overhead, the security level of access control in ICN architectures is remarkably increased.

6.5 Analysis of ECAC Attributes

In this subsection, we evaluate ECAC with respect to the eight attributes of a good access control mechanism (see the attribute list in Section 6.1). The communication overhead is only extra five messages in ECAC. The existing protocols are using at least eight messages. There is no required modification in the ICN architectures for our proposed access control protocol. There is no need to add any extra entities. All the modifications are in the public messages or are extra computations done by ICN entities. This covers the reduction of unnecessary communication overhead and the elimination of modifications to ICN architecture attributes of a good access control mechanism.

The protocol is based on one shared key that needs to be exchanged in ECAC protocol. The other secret parameters (k_1, k_2) do not need to be exchanged. For ICN users, each one has a public and private key pair. In the proposed protocol, we have included hashing calculations and hashing verifications. Also, we have added encryption for transmitted messages, random number generations, and calculations for shared secret key. This satisfies the minimization of the exchange of secret keys and extra operations for access control attributes of a good access control mechanism.

The protocol prevents man-in-the-middle attacks, as an attacker cannot impersonate an authenticated user or ICN node because the attacker needs to know secret information that cannot be easily compromised using the public messages in ECAC. Additionally, the protocol prevents forward security and replay attacks. This achieves the fifth attribute of a good access control mechanism.

The privacy attribute has been achieved by using elliptic curve cryptography, shared keys, and nonces. Information that can lead to privacy violations in ECAC are encrypted using receiver's public key in addition to the random number selected by the sender, which means that no one can decrypt it except the intended receiver. Other ICN privacy related issues are left as a future work. By comparing the two hash values, ICN nodes are able to check the integrity of ICN subscriber's requests. By comparing the two keys, in addition to self-certifying verifications, ICN subscribers are able to check the integrity and authenticity of ICN contents. This fulfills the sixth and seventh attributes of a good access control mechanism.

For the ICN naming scheme, we use the self-certifying naming scheme in ECAC because it is the most popular ICN naming scheme used in many ICN architectures. The related access control protocols mentioned in these papers [11, 35, 80] also use self-certifying

naming scheme. Hierarchal naming scheme needs additional security add-ons to achieve content authenticity and integrity.

6.6 Comparison with ICN Access Control Protocols

In addition to the related work discussed in Section 6.2, we make a direct comparison between ECAC and the related centralized and decentralized access control protocols. As shown in Table 6.3, we use the following criteria as a basis for comparison: number of public authentication messages, number of extra entities, number of secret keys, number of ICN names for each content, required security techniques, authentication type, access control decision, and ICN architecture.

Table 6.3 indicates that DACPI and ECAC have the minimum number of public messages and do not require any additional modifications to the architecture. In [35], eight public messages are needed to enforce the access control for both publication and subscription functions. In [11], a total of eight messages are needed for access control distributed as four messages for publication and four for subscription. In [35], two extra entities are needed, while in [11], one extra entity is needed for access control purposes. In SAC [80], a total of eleven messages are needed for publication and subscription. SAC does not need extra entities to ICN architectures.

For the required secret information, DACPI and ECAC use the same or fewer secret keys than the other protocols. In both protocols, we need one shared key and each user has a public and a private key. ECAC uses smaller key size to achieve the same security level of DACPI. All four access control mechanisms use only one name for each ICN content except the decentralized mechanism [80] which uses two names for each content. The secure name in [80] will be known to authenticated users only. DACPI uses extra cryptographic

techniques with respect to the other protocols, while ECAC uses less cryptographic techniques than DACPI and similar to the decentralized mechanism [80]. Although the number of parameters in ECAC is bigger than DAPCI, the required space in ECAC is less because elliptic curve cryptography can achieve similar security levels with smaller key sizes relative to RSA. We use key size of 160 bits in ECAC to have the same security level of 1024 bits key size in DACPI. ECAC provides faster computations because messages generated from ECAC are shorter and require less space than DACPI. These advantages enable ECAC to decrease these extra overhead for ICN architectures.

Fotiou et al. [35] design their solution for publish subscribe Internet technology (PURSUIT) ICN architecture, while Aiash et al. [11] propose their solution for network of information (NetInf) ICN architecture. The other three decentralized mechanisms can be applied to different ICN architectures. Centralized access control mechanisms need to be bounded to a specific architecture because they need architectural modification by adding extra entities, while decentralized ones can be adapted to be implemented in different architectures. This is a major advantage of decentralized schemes over the centralized ones in ICN architectures.

In terms of the impact on ICN performance, ECAC achieves results similar to centralized access control scheme representative (ACED) without the need of adding new entities or architectural modification to ICN architecture.

As mentioned in the related work section, many works use encryption-based techniques for achieving access control. Encryption-based access control techniques require either modifications to contents or ICN requests to encrypt the transmitted data. Additionally, encryption-based techniques require high computational overhead, because they need to encrypt high volumes of contents or requests. ECAC depends on one of the existing ICN

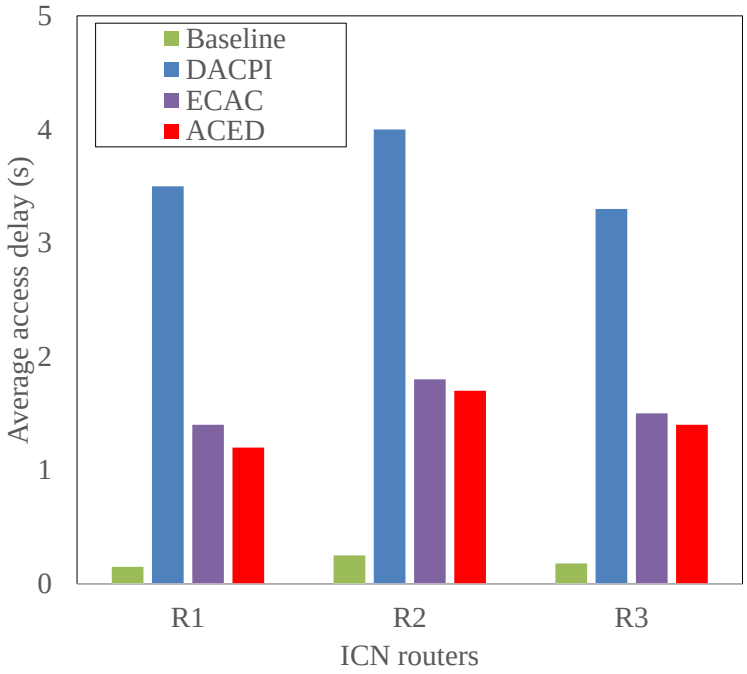

Figure 6.8: ICN average request access time delay at three edge routers for baseline, DACPI, ECAC, and ACED protocols with request rate 20 req/sec with cache size= 5000 entry

naming schemes instead of proposing a new or modified version of ICN naming or requests. In ECAC, we do not encrypt contents or subscription requests, we exchange them as proposed by ICN architectures. We only encrypt and secure the parameters required for access control purposes.

6.7 Evaluation of Access Time Delay

In this section, we evaluate the impact of ECAC on ICN performance. We measure access time delay without ECAC protocol and related access control protocols and in the existence of these protocols. Request access time delay is measured for different request rates, cache sizes, and various ratios of attackers to legitimate users.

Table 6.3: Comparison between ECAC and existing centralized and decentralized access control protocols

	Access control enforcement delegation (ACED) [35]	Authentication and authorization approach [11]	Session-based authentication (SAC) [80]	Decentralized access control protocol for ICN (DACPI) [6]	Elliptic curve based access control for ICN (ECAC)
Number of public authentication messages	Eight for both subscription and publication	Four for subscription and four for publication	Six for subscription and five for publication	Five for both subscription and publication	Five for both subscription and publication
Number of extra entities	Two (AC provider and relaying party)	One (trusted ticket granting)	Zero	Zero	Zero
Number of secret keys	Two secret keys one for publisher and one for subscriber	Three secret keys and each publisher and NRS has public and private keys	Three symmetric keys and each content provider has public and private keys	One key and each user has public and private keys	One secret key and each user has public and private keys. Other secrets do not need to be exchanged.
Number of ICN names for each content	One	One	Two (public and secure name)	One	One
Required security techniques	No extra techniques are required	Public key infrastructure and hashing	Public key infrastructure, hashing, and random number generation	RSA, hashing, Diffie-Hellman, and random number generation	Elliptic curve, hashing, and random number generation
Authentication type	One-way (access control provider authenticates ICN)	One-way (name resolution service authenticates ICN)	Mutual authentication between ICN nodes and ICN users	Mutual authentication between ICN nodes and ICN users	Mutual authentication between ICN nodes and ICN users
Access control	Centralized	Centralized	Decentralized	Decentralized	Decentralized
ICN architecture	Publish Subscribe Internet Technology (PURSUIT)	Network of Information (NetInf)	Can be applied to different ICN architectures	Can be applied to different ICN architectures	Can be applied to different ICN architectures

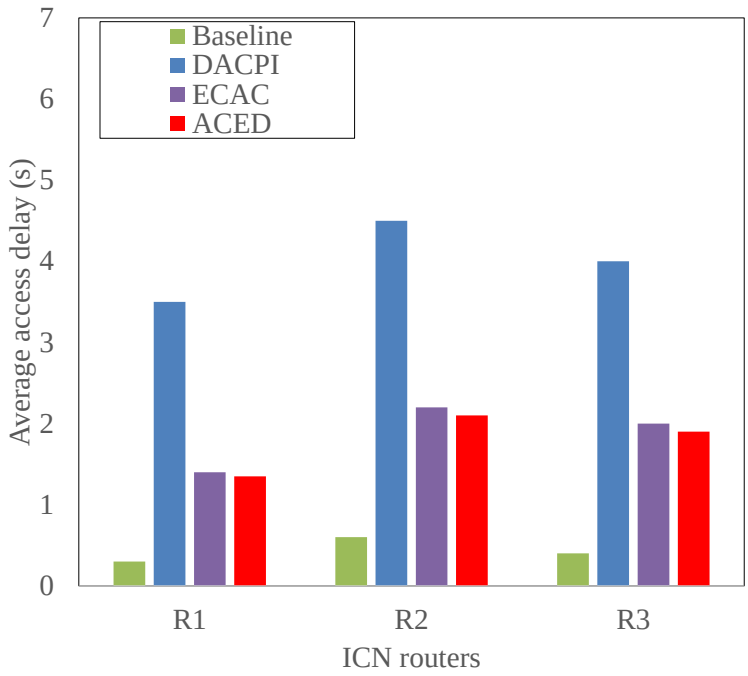

Figure 6.9: ICN average request access time delay at three edge routers for baseline, DACPI, ECAC, and ACED protocols with request rate 100 req/sec with cache size= 5000 entry

6.7.1 Unauthorized Access Attack Experiments

Access time delay in ICN is a major issue because of the ICN property that each request has one response and there is no response without a request. In non-ICN architectures, requests can receive many data packets. This means that security solutions are applied on each request. Our objective is to achieve access time delay results in ICN architectures close to centralized access control mechanisms. Centralized access control mechanisms add extra entities to manage access control policies to reduce required overhead. Experimental results show that ECAC achieves results similar to the ACED in all cases. In the experiments, we evaluate the average access time delay caused by our proposed protocol and compare it with related protocols under different scenarios. We study the impact of our proposed protocol

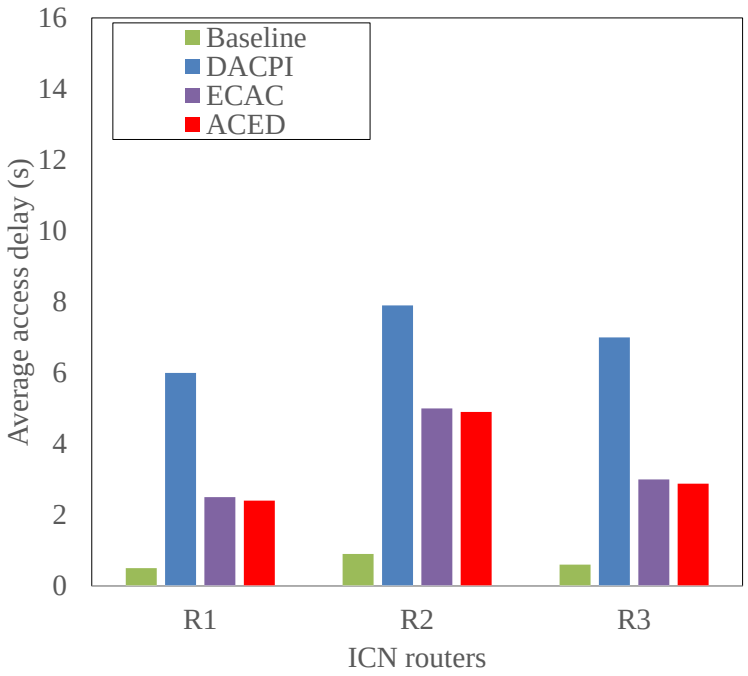

Figure 6.10: ICN average request access time delay at three edge routers for baseline, DACPI, ECAC, and ACED protocols with request rate 200 req/sec with cache size= 5000 entry

on ICN by measuring the average request access time delay, which represents the delay between the first request sent and data packet received. We evaluate ECAC using ndnSIM, which is a simulator for the NDN architecture within NS-3 [9]. NDN architecture uses the hierarchal naming scheme. We use the simulator for checking ICN performance with and without the access control schemes.

We build our experiments using backbone AT&T network, which is Internet-like architecture. In our experiments, the network consists of 150 subscribers, ten publishers and more than 40 routers as shown in Figure 4.1. Each edge router is connected to ten ICN users. We change the subscriber's request rates between 20, 100, and 200 requests per second and cache size between 1000 and 5000 entries. We also use the following parameters: no. of Pending Interest Table (PIT) entries = 1000 entry, payload size = 1 KB. We use

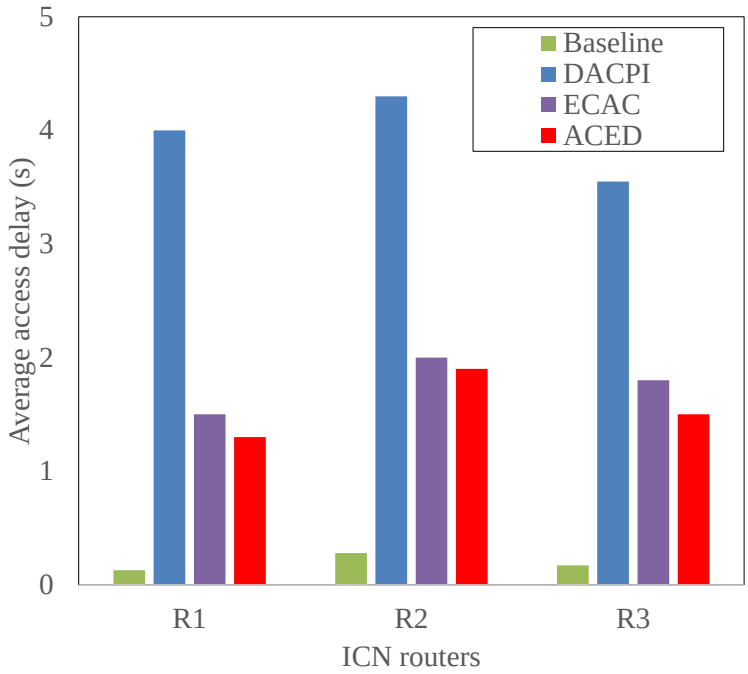

Figure 6.11: ICN average request access time delay at three edge routers for baseline, DACPI, ECAC, and ACED protocols with request rate 20 req/sec with cache size= 1000 entry

default parameters for point to point links and channels as follows: point to point channel data rate = 1 Mbps, point to point channel delay = 10 ms. We set key size for DACPI = 1024 bits, Key size for ECAC = 160 bits, Key size for ACED = 80 bits. These key sizes achieve the same security level. As a caching replacement strategy, we use least frequently used technique to evict the least popular contents.

We compare the average request access time delay between a baseline case, DACPI, ECAC, and ACED. In the baseline, we measure ICN request access time delay when there is no add-on solution. DACPI is chosen as a representative of decentralized access control mechanisms for ICN. ACED is selected as a representative for centralized access control mechanisms for ICN. In Figure 6.8 - Figure 6.13, results are obtained for different request rates (20, 100, and 200 requests/sec) and for different cache sizes (5000 and 1000 entry). In

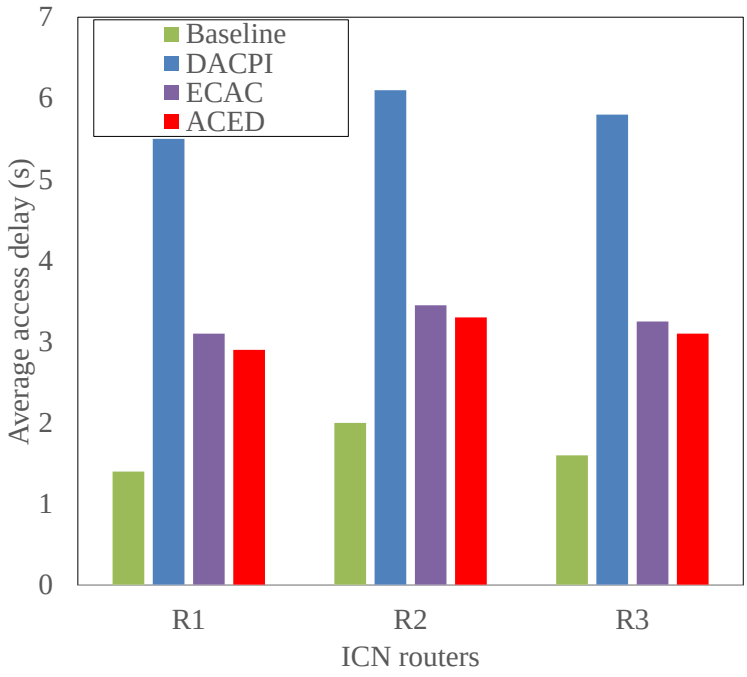

Figure 6.12: ICN average request access time delay at three edge routers for baseline, DACPI, ECAC, and ACED protocols with request rate 100 req/sec with cache size= 1000 entry

Figure 6.14 - Figure 6.16, results are obtained for different ratios of attackers to legitimate users (20%, 50%, and 80%) with the same request rates for users and attackers, which is 100 requests/sec. Results are recorded at three random edge routers as examples and similar results can be obtained from other routers.

6.7.2 Discussion

The experimental results show that the three access control mechanisms increase request access time delay in all cases because of the extra time needed for enforcing access control in ICN. This extra time is coming from either the exchange of access control messages between different entities as in ACED or applying the cryptographic techniques as in ECAC

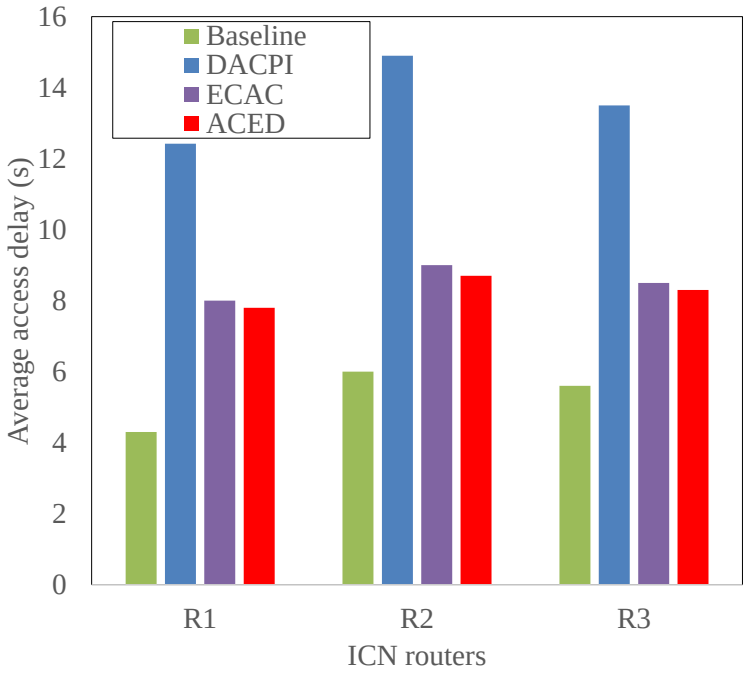

Figure 6.13: ICN average request access time delay at three edge routers for baseline, DACPI, ECAC, and ACED protocols with request rate 200 req/sec with cache size= 1000 entry

and DACPI. ECAC and ACED achieve similar results in all cases and outperform DACPI. The reason behind these almost similar results between ECAC and ACED comes from the difference between the number of messages required for access control. Although in ACED, there is no encryption and decryption as in ECAC, the extra messages and exchange of these messages between the extra entities in ACED minimize the time difference between the two protocols. Based on Figure 6.8 - Figure 6.13, as the cache size increases, request access time delay for all cases and the difference between the three access control protocols and baseline decrease. It happens because the number of cache content evictions decreases as the cache size increases, and hence routers can use their local copies instead of sending requests to the original sources.

As depicted in Figure 6.8, the average request access time delay is recorded as 0.2, 3.6,

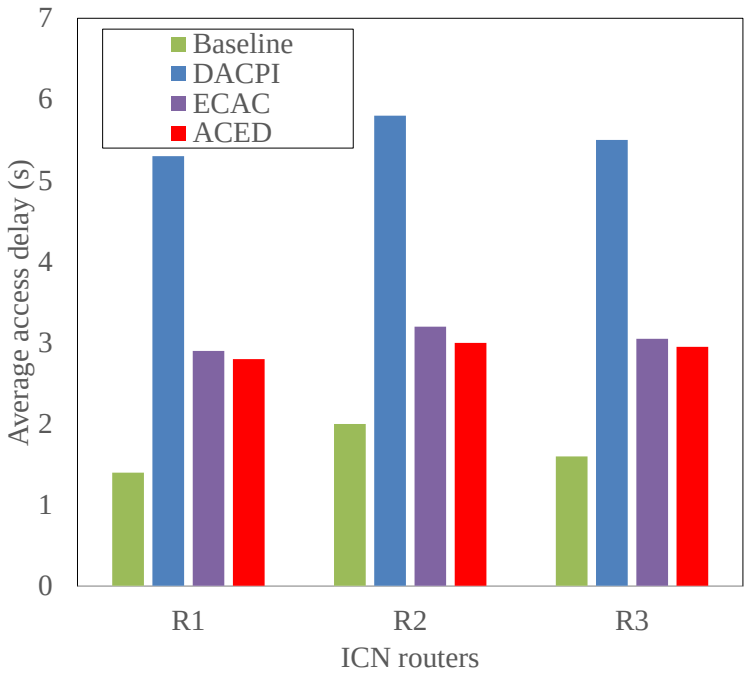

Figure 6.14: ICN average request access time delay at three edge routers for baseline, DACPI, ECAC, and ACED protocols with 20% attackers exist, cache size= 1000 entry and request rate= 100 req/sec

1.5, and 1.4 for the baseline, DACPI, ECAC, and ACED, respectively. When 100 requests per second with cache size of 5000 entry, as shown in Figure 6.9, the values of average access delay become 0.4, 4, 1.8, and 1.7 for the same protocols. The average access time delay is 0.6 seconds for the baseline, 6.9 seconds for DACPI, 3.5 seconds for ECAC, and 3.3 seconds for ACED, as in the last subcase as depicted in Figure 6.10.

As the number of requests increases, the required access time delay gradually increases. In Figure 6.11, the average access time delay is 0.2, 4, 1.7, and 1.5 seconds for the baseline, DACPI, ECAC, and ACED, respectively. The average access time delay is 1.6 seconds for the baseline, 5.8 seconds for DACPI, 3.2 seconds for ECAC, and 3.1 seconds for ACED, as depicted in Figure 6.12. Figure 6.13 shows that the average access time delay is 5.3, 13.6, 8.5, and 8.2 seconds for the same protocols.

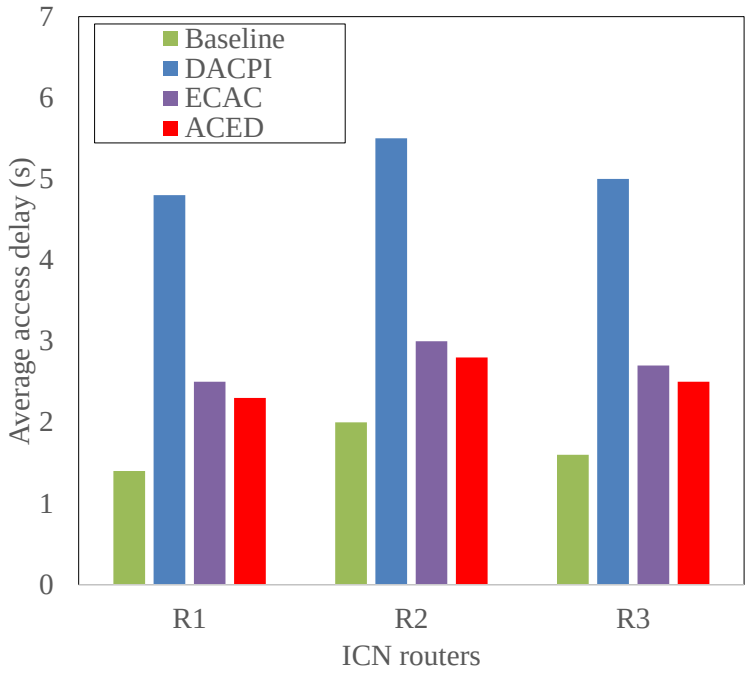

Figure 6.15: ICN average request access time delay at three edge routers for baseline, DACPI, ECAC, and ACED protocols 50% attackers exist, cache size= 1000 entry and request rate= 100 req/sec

In the normal case, when there is no attack as shown in Figure 6.11 - Figure 6.13, the differences between the baseline and the three access control mechanisms are more than the differences in the attack case. Also, as the number of attackers increases, the three protocols achieve better results with respect to the baseline, as shown in Figure 6.14 - Figure 6.16. In the attack case, the three protocols succeed to detect attackers and reduce access control messages and related computations. Attackers send uniformly distributed malicious requests, which means the same number of requests for each content. When 20% attackers exist as shown in Figure 6.14, the average access time delay is 1.6, 5.5, 3, and 2.9 seconds for baseline, DACPI, ECAC, and ACED, respectively. When 50% attackers exist as depicted in Figure 6.15, the average access time delay is 1.6, 5.1, 2.7, and 2.5 seconds for the same protocols. The average access time delay is 1.6 seconds for the baseline,

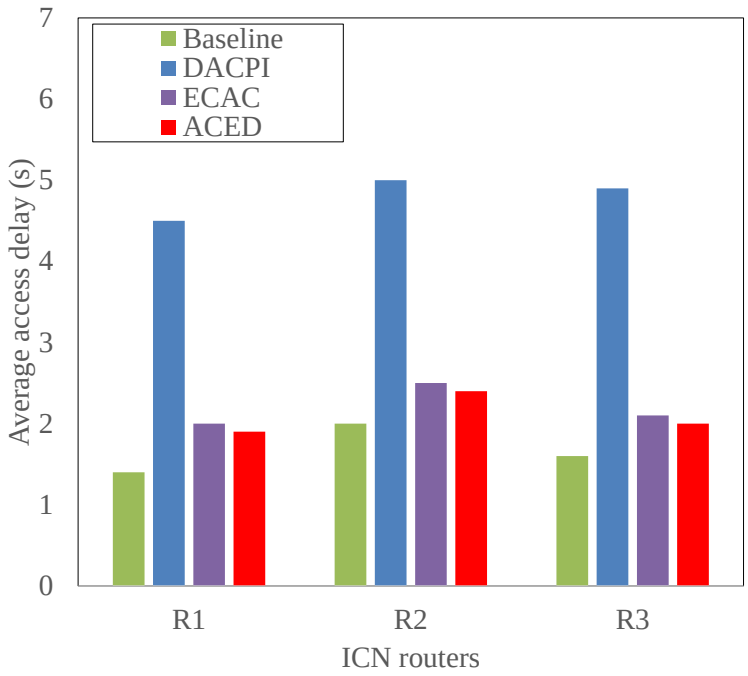

Figure 6.16: ICN average request access time delay at three edge routers for baseline, DACPI, ECAC, and ACED protocols 80% attackers exist, cache size= 1000 entry and request rate= 100 req/sec

4.8 seconds for DACPI, 2.2 seconds for ECAC, and 2.1 seconds for ACED, as shown in Figure 6.16 when 80% attackers exist.

6.8 Summary

In ICN, access control enforcement becomes a major problem because contents are highly distributed and cached everywhere in ICN architectures and subscribers can access these contents from any available copy. ICN requires an access control solution that should be integrated with the architecture itself.

In this chapter, we first propose a decentralized access control protocol (DACPI) depending on RSA and Diffie-Hellman cryptographic techniques. We then propose an enhanced protocol using elliptic curve cryptography (ECAC) to achieve access control security service to ICN contents. The security of elliptic curve cryptography depends on the difficulty of solving the elliptic curve logarithm problem. ECAC depends on elliptic curve, hashing, and random number generations. ECAC protocol is a decentralized access control mechanism, which does not require new entities to ICN architectures. Access control decisions in ECAC are taken by ICN nodes and subscribers themselves without a central authority. Using ECAC, only authenticated users can access authenticated contents.

ECAC successfully prevents man-in-the-middle, forward security, replay attacks, integrity, and privacy attacks related to the access control process. The main idea is that an attacker cannot retrieve the secret keys based on public parameters used in the proposed protocols. The attacker cannot form valid authentication messages without knowing the secret keys. ECAC also efficiently minimizes the number of public messages and storage overhead for the access control process. With the light communication, computational, and storage overhead used in ECAC, the access control security level is remarkably increased. Based on our experiments, ECAC uses fewer public messages and does not require extra entities as in the centralized mechanisms. To achieve access control security service in ICN, we highly recommend using decentralized access control based on elliptic curve cryptography. This option can achieve the best performance results without architectural modifications.

Chapter 7

Conclusions and Future Directions

7.1 Summary

In this thesis work, our goal was to develop a security framework for ICN architectures. Our framework is integrated within the architecture that delivers contents with high availability and in a secure manner to legitimate users. Additionally, the framework preserves the privacy of ICN users and contents.

Chapter 2 discussed the need for new architectures for Next Generation Internet (NGI) and basic ICN components and architectures. We highlighted the unique characteristics of ICN.

Chapter 3 presented a taxonomy of ICN attacks to classifiy the attacks into naming, routing, caching, and miscellaneous attacks. These attacks have different impacts on unique ICN attributes: location independent naming, state decorrelation, in-network caching, and ubiquitous publication/subscription. ICN attacks also have various influences on availability, access control, and privacy. Based on ten evaluation metrics, we calculated the severity levels of ICN attacks which can be classified into low, medium, and high levels.

Chapter 4 described our reference model that consists of ICN routers, distributed storage location, subscribers, and publishers. We used three indicative performance metrics: satisfied legitimate requests, cache hit ratio, and request access time delay. These performance metrics indicated the effectiveness of our proposed solutions with respect to baseline and other solutions. Additionally, we presented the proposed framework including the required functions in the availability, access control, and privacy components. This framework will be useful for traffic management to be implemented in ICN architectures.

ICN is an open environment that depends on in-network caching and focuses on contents. These attributes make ICN architectures subject to different types of routing and caching attacks. An attacker publishes invalid contents or announces malicious routes and sends malicious requests for available and unavailable contents. These types of attacks can cause Distributed Denial of Service (DDoS) and cache pollution in ICN architectures. **Chapter 5** proposed Defending Against DDoS in ICN routing and caching (DADI) solution. This solution allows ICN routers to differentiate between legitimate and attack behaviors in the detection phase based on threshold values. In the prevention phase, ICN routers are able to take actions against these attacks. In our experiments, we measured satisfied requests for legitimate users and cache hit ratio for ICN routers, which are evaluated over different scenarios when there are 20%, 50%, and 80% attackers with respect to legitimate users. The experiments demonstrate that the proposed solution effectively mitigates routing and caching related DDoS attacks in ICN and enhances ICN performance in the existence of DDoS attacks.

ICN has in-network caching capability, which enables any node to cache any content coming from any publisher. ICN subscribers can access contents from different distributed

locations. This in-network caching capability maximizes the problem of unauthorized access to ICN contents.

Chapter 6 first proposed a decentralized access control protocol (DACPI) depending on RSA and Diffie-Hellman cryptographic techniques and then proposed a decentralized Elliptic Curve based Access Control (ECAC) protocol for ICN architectures. In ECAC, fewer public messages are needed for access control enforcement between ICN subscribers and ICN nodes than the existing access control protocols. ECAC protocol depends on ICN self-certifying naming scheme. We performed security analysis on ECAC for the following attacks: man-in-the-middle, forward security, replay, integrity, and privacy violations. We also evaluated communication, computational, and storage overhead for ECAC. According to the security and performance analysis, ECAC prevents unauthorized access to ICN contents with a minimum overhead.

7.2 Limitations and Future Directions

Our proposed solution for ICN routing and caching DDoS related attacks (DADI) has some limitations that can be summarized in the following points. An attacker can send malicious requests within the bounds of legitimate users. DADI does not achieve the same performance in all attack cases and sending malicious uniform requests is considered to be the worst case. There are other naming DDoS related attacks that can happen and are not covered by DADI. ECAC also has some limitations. ECAC cannot be applied with hierarchical naming scheme because it depends on self-certifying naming scheme authenticity and integrity verification for contents. ECAC is applied for each request due to the ICN nature that each request is satisfied individually. ECAC cannot be applied at once on all requests asking for the whole content.

The main threats to validity of this thesis research can be summarized as follows. First, we did not assume the existence of malicious requests while calculating the threshold values of request satisfaction ratio (RSR), request rate, rating for contents, rating for publishers, and rating for cached contents. Second, ICN routers can be compromised by inside or outside attackers. With respect to the routers, outside attackers represent malicious publishers or subscribers, while the inside attackers can modify the router software or hardware to launch attacks by employing those routers. We provide solutions against the attacks generated only by malicious publishers and subscribers.

Several future directions and open issues stem from our work. We point out some research issues related to the proposed framework components and their integration.

Privacy component. This component prevents the privacy related attacks, which allow an attacker to gain private information about ICN users and contents. The privacy related attacks cover the naming, interception, and time analysis attacks. The naming related attacks can be classified into watchlist and sniffing attacks. In interception attacks, an attacker masquerades as a trusted publisher and announces invalid routes, while maintaining a record of valid routes to the content. Content requests can then be captured and sent to the proper location. Although the receiver gets the content normally, the attacker gains knowledge of the requested content. Existing solutions for naming related attacks, such as mix-nets [23], Tor [31], Freedom [97], Anonymizer [95], Freenet [26], and deniable encryption [19], cannot be applied in ICN as they are not designed for environments in which content is the main focus to the architecture. The ICN security solution should achieve privacy, censorship resistance and plausible deniability for users. The solution should also be computationally easy for the user to retrieve the content and computationally expensive for the attacker to identify or detect the name requested or content retrieved. The objective in this component

is to preserve the privacy for both ICN users and their contents. There are other solutions designed for the smart grid data collection and web applications that can be investigated for the ICN architectures [16, 48, 49, 51, 71, 79].

Mobility attacks. In ICN, mobile attackers publish and subscribe contents to a region by traversing neighboring networks on circular paths whereas sending an enormous number of content requests. The attacker aims to overload the mobile access routers to make it exceed the state timeout, which leads to a blockade of the available networks. In an ICN architecture, the retransmission of requests is part of the mobility nature, which adds difficulty in detecting this attack [77]. The difficulty in ICN is to identify or detect these mobile attackers.

Integration with non-ICN architectures. We believe that ICN will be incrementally deployed with non-ICN architectures. The future work stemming from this thesis is to have an interface to connect ICN with non-ICN architectures. The interface should include the appropriate functions and parameters that allow contents to be transferred properly and securely between ICN and non-ICN architectures.

Bibliography

[1] E. G. AbdAllah, H. S. Hassanein, and M. Zulkernine, "A survey of security attacks in information-centric networking", *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, 2015, pp. 1441-1454.

[2] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "A security framework for ICN traffic management", *Submitted to IEEE Communications Magazine - Feature Topic on Traffic Measurements for Cyber Security, October 2016*.

[3] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "Countermeasures for mitigating ICN routing related DDoS attacks", *The 10th Int. Conf. on Security and Privacy in Communication Networks (Securecomm14)*, Beijing, China, 2014, pp. 84-92.

[4] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "Detection and prevention of malicious requests in ICN routing and caching", *The 13th IEEE Int. Conf. on Dependable, Autonomic, and Secure Computing (DASC-2015)*, Liverpool, UK, 2015, pp. 1741-1748.

[5] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "DADI: defending against DDoS in ICN routing and caching", *Submitted to IEEE Transactions on Dependable and Secure Computing (TDSC), March 2016*.

[6] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "DACPI: A decentralized access control protocol for information centric networking", *IEEE Symposium on Communication and Information System Security - ICC'16*, to appear, Kuala Lumpur, Malaysia, May 2016, DOI: 10.1109/ICC.2016.7511198.

[7] E. G. AbdAllah, M. Zulkernine, and H. S. Hassanein, "Preventing unauthorized access in information centric networking", *Submitted to Elsevier Journal of Network and Computer Applications, January 2017*.

[8] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest flooding attack and countermeasures in named data networking", *IFIP Networking Conference*, 2013, pp. 1-9.

[9] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3", NDN, Technical Report NDN-0005, 2012, [Online]. Available: `http://named-data.net/wp-content/uploads/TRndnsim.pdf`, accessed on 23 March, 2017.

[10] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking", *IEEE Communications Magazine*, vol. 50, no. 7, July 2012, pp. 26-36.

[11] M. Aiash and J. Loo, "An integrated authentication and authorization approach for the network of information architecture", *Journal of Network and Computer Applications*, vol. 50, 2014, pp. 73-79.

[12] F. Almeida and J. Loureno, "Information centric networks-design issues, principles and approaches", *International Journal of Latest Trends in Computing*, vol. 3, no. 3, 2012, pp. 58-66.

[13] S. Arianfar, T. Koponen, B. Raghavan, and S. Shenker, "On preserving privacy in content-oriented networks", *ACM SIGCOMM Workshop on Information-Centric Networking ICN'11*, August 2011, pp. 19-24.

[14] R. Azarderakhsh, K. U. Jarvinen, and M. Mozaffari-Kermani, "Efficient algorithm and architecture for elliptic curve cryptography for extremely constrained secure applications", *IEEE Transactions on Circuits and Systems*, vol. 61, no. 4, 2014, pp. 1144-1155.

[15] Md. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu, "A survey of naming and routing in information-centric networks", *IEEE Communications Magazine*, vol. 49, no. 12, 2012, pp. 44-53.

[16] A. Barua, H. Shahriar, and M. Zulkernine, "Server-side detection of content sniffing attacks", *Proc. of the 22nd Annual International Symposium on Software Reliability Engineering (ISSRE)*, Hiroshima, Japan, November 2011, pp. 20-29.

[17] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: evidence and implications", *IEEE Infocom'09*, New York, USA, 1999, pp. 1-9.

[18] J. Burke, A. Horn, and A. Marianantoni, "Authenticated lighting control using named data networking", *NDN Technical Report NDN-0011*, University of California, Los Angeles, 2012, [Online]. Available: `http://named-data.net/`

wp-content/uploads/TRlighting.pdf, accessed on 23 March, 2017.

[19] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable encryption", *in Proc. of CRYPTO*, Lecture Notes in Computer Science, vol. 1294, 1997, pp. 90-104.

[20] G. Carofiglioa, G. Morabitob, L. Muscarielloc, I. Solisd, and M. Varvelloe, "From content delivery today to information centric networking", *Computer Networks*, vol. 57, no. 16, 2013, pp. 3116-3127.

[21] A. Chaabane, E. De Cristofaro, M. A. Kaafar, and E. Uzun, "Privacy in content-oriented networking: threats and countermeasures", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 43, no. 3, 2013, pp. 25-33.

[22] H. Chabanne and M. Tibouchi, "Securing e-passports with elliptic curves", *IEEE Security & Privacy Magazine*, vol. 9, no. 2, 2011, pp. 75-78.

[23] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms", *Communications of the ACM*, vol. 24, no. 2, 1981, pp.84-90.

[24] D. Cheriton and M. Gritter, "TRIAD: a scalable deployable NATbased Internet architecture", Technical Report, January 2000. [Online]. Available: https://www.cs.tcd.ie/~htewari/papers/triad.pdf, accessed on 23 March, 2017.

[25] C. Chung, P. Khatkar, T. Xing, J. Lee, and D. Huang, "NICE: network intrusion detection and countermeasure selection in virtual network systems", *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 4, June 2013, pp. 198-211.

[26] I. Clarke, T. W. Hong, S. G. Miller, O. Sandberg, and B. Wiley, "Protecting free expression online with freenet", *IEEE Internet Computing*, vol. 6, no. 1, 2002, pp. 40-49.

[27] A. Compagno, M. Conti, P. Gasti, and G. Tsudik, "Poseidon: mitigating interest flooding DDoS attacks in named data networking", *arXiV:1303.4823v3 [cs.NI]*, August 2013, [Online]. Available: `http://arxiv.org/pdf/1303.4823.pdf`, accessed on 23 March, 2017.

[28] M. Conti, P. Gasti, and M. Teoli, "A lightweight mechanism for detection of cache pollution attacks in named data networking", *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 57, no. 16, November 2013, pp. 3178-3191.

[29] C. Dannewitz, J. Golic, B. Ohlman, and B. Ahlgren, "Secure naming for a network of information", *in Proc. of the IEEE Infocom*, March 2010, pp. 1-6.

[30] M. D'Ambrosio, C. Dannewitz, H. Karl, and V. Vercellone, "MDHT: a hierarchical name resolution service for information-centric networks", *in Proc. ACM SIGCOMM Workshop on ICN*, 2011, pp. 7-12.

[31] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router", *in Proc. of the 13th USENIX Security Symposium*, 2004.

[32] D. Djenouri, L. Khelladi, and A. Badache, "A survey of security issues in mobile ad hoc and sensor networks", *IEEE Communications Surveys & Tutorials*, vol. 7, no. 4, 2005, pp. 2-28.

[33] X. Du, M. Guizani, Y. Xiao, and H. H. Chen, "A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks", *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, 2009, pp. 1223-1229.

[34] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Fighting spam in publish/subscribe networks using information ranking", *Next Generation Internet (NGI), 6th EURO-NF Conference*, Paris, June 2010, pp. 1-6.

[35] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Access control enforcement delegation for information-centric networking architectures", *The Second Edition of the ICN Workshop on Information-Centric Networking*, ACM, 2012, pp. 85-90.

[36] Y. Gao, L. Deng, A. Kuzmanovic, and Y. Chen, "Internet cache pollution attacks and countermeasures", *IEEE International Conference on Network Protocols - ICNP*, November 2006, pp. 54–64.

[37] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "DoS & DDoS in nameddata networking", *in Proc. of the 22nd International Conference on Computing Communications and Networks (ICCCN)*, Nassau, August 2013, pp. 1-7.

[38] C. Ghali, M. A. Schlosberg, G. Tsudik, and C. A. Wood, "Interest-based access control for content centric networks (extended version)", *ICN'15*, San Francisco, CA, USA, 2015, pp. 1-10.

[39] C. Ghali, G. Tsudik, and E. Uzun, "Needle in a haystack: mitigating content poisoning in named-data networking", *SENT'14*, San Diago, CA, USA, 2014.

[40] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric networking: seeing the forest for the trees", *in Proc. of the 10th ACM Workshop on Hot Topics in Networks*, 2011, pp. 1-6.

[41] L. Guo, E. Tan, S. Chen, Z. Xiao, and X. Zhang, "Does internet media traffic really follow Zipf-like distribution?", *ACM Sigmetrics International Conference on Measurement and Modeling of Computer Systems*, New York, USA, 2007, pp. 359-360.

[42] S. Hansman and H. Ray, "A Taxonomy of network and computer attacks", *Computers & Security*, vol. 24, no. 1, 2005, pp. 31-43.

[43] D. He and S. Zeadally, "An analysis of RFID authentication schemes for Internet of things in healthcare environment using elliptic curve cryptography", *IEEE Internet of Things Journal*, vol. 2, no. 1, 2015, pp. 72-83.

[44] V. M. Igure and R. D. Williams, "Taxonomies of attacks and vulnerabilites in computer systems", *IEEE Communications Surveys & Tutorials*, vol. 10, no. 1, 2008, pp. 6-19.

[45] M. Ion, J. Zhang, and E. M. Schooler, "Toward content-centric privacy in ICN: attribute-based encryption and routing", *ASIA'CCS13*, Hangzhou, China, 2013, pp. 513-514.

[46] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. Braynard, "Networking named content", *CoNEXT'09*, ACM, December 2009, pp. 1-12.

[47] A. Keromytis, V. Misra, and D. Rubenstein, "SOS: An Architecture for mitigating DDoS attacks", *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, 2004, pp. 176-188.

[48] Y. Kim, V. Kolesnikov, H. Kim, and M. Thottan, "SSTP: a scalable and secure transport protocol for smart grid data collection", *IEEE International Conference on for Smart Grids Communications (SmartGridComm)*, October 2011, pp. 161-166.

[49] Y. Kim, V. Kolesnikov, and M. Thottan, "Resilient end-to-end message protection for large-scale cyber-physical system communications", *IEEE Third International Conference on for Smart Grids Communications (SmartGridComm)*, November 2012, pp. 193-198.

[50] Y. Kim, W. C. Lau, M. C. Chuah, and H. J. Chao, "PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service attacks", *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 2, May 2006, pp. 141-155.

[51] Y. Kim, J. Lee, G. Atkinson, H. Kim, and M. Thottan, "SeDAX: a scalable, resilient, and secure platform for smart grid communications", *IEEE Journal on Selected Areas in Communications*, July 2012, pp. 1119-1136.

[52] E. Kline, A. Afanasyev, and P. Reiher, "Shield: DoS filtering using traffic deflecting", *19th IEEE International Conference on Network Protocols*, 2011, pp. 37-42.

[53] T. Koponen, M. Chawla, B. G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture", *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, 2007, pp. 181-192.

[54] J. Kurihara, E. Uzun, and C. A. Wood, "An encryption-based access control framework for content-centric networking", *IFIP Networking*, 2015, [Online]. Available: `https://www.ietf.org/mail-archive/web/icnrg/current/pdf9_JIQ5GBeS.pdf`, accessed on 23 March, 2017.

[55] D. Lagutin, K. Visala, and S. Tarkoma, "Publish/subscribe for internet: PSIRP perspective", *Towards the future Internet*, IOS Press, vol. 4, 2010, pp. 75-84.

[56] Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede, "Elliptic-curve-based security processor for RFID", *IEEE Transactions on Computers*, vol. 57, no. 11, 2008, pp. 1514-1527.

[57] L. Lu, M. C. Chan, and E. C. Chang, "A general model of probabilistic packet marking for ip traceback", *ASIA CCS*, 2008, pp. 179-188.

[58] E. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes", *IEEE Communications Surveys & Tutorials*, vol. 7, no. 2, 2005, pp. 72-93.

[59] K. Maletsky, "RSA vs ECC comparison for embedded systems", *Atmel*, 2015, [Online]. Available: `http://www.atmel.com/images/atmel-8951-cryptoauth-rsa-ecc-comparison-embedded-systemcs-whitepaper.pdf`, accessed on 23 March, 2017.

[60] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS attack and DDoS defense mechanisms", *ACM*, 2004.

[61] J. Mirkovic and P. Reiher, "D-WARD: a source-end defense against flooding denial-of-service attacks", *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 3, September 2005, pp. 216-232.

[62] S. Misra, R. Tourani, and N. E. Majd, "Secure content delivery in information centric networks: design, implementation, and analyses", *ICN*, Hong Kong, China, 2013, pp. 73-78.

[63] A. Mohaisen, H. Mekky, X. Zhang, H. Xie, and Y. Kim, "Timing attacks on access privacy in information centric networks and countermeasures", *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 6, November 2015, pp. 675-687.

[64] A. Mohaisen, X. Zhang, M. Schuchard, H. Xie, and Y. Kim, "Protecting access privacy of cached contents in information centric networks", *SIGCOMM'13*, Hong, Kong, China, May 2013, pp. 1001-1003.

[65] H. Moustafa and S. Zeadally, "Media networks: architectures, applications and standards", *CRC Press*, 2012.

[66] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G.Pantziou, "A survey on jamming attacks and countermeasures in WSNs", *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, 2009, pp. 42-56.

[67] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures", *IEEE Communications Magazine*, vol. 49, no. 7, July 2011, pp. 26-36.

[68] A. M. K. Pathan and B. Rajkumar, "A taxonomy and survey of content delivery networks", Grid Computing and Distributed Systems Laboratory, University of Melbourne, Technical Report, 2007.

[69] M. L. Polla, F. Martinelli, and D. Sgandurra, "A survey on security for mobile devices", *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, 2012, pp. 446-471.

[70] R. Rembarz, D. Catrein, and J. Sachs, "Private domains in networks of information", *IEEE International Conference on Communications Workshops*, Dresden, June 2009, pp. 1-5.

[71] H. Shahriar and M. Zulkernine, "Client-side detection of cross-site request forgery attacks", *Proc. of the 21st IEEE International Symposium on Software Reliability Engineering (ISSRE)*, San Jose, USA, November 2010, pp. 358-367.

[72] D. K. Smetters, P. Golle, and J. D. Thornton, "CCNx access control specications", *PARC*, Tech. Rep, 2010, [Online]. Available: `http://www.arl.wustl.edu/~jdd/NDN/CCNX/ccnx/doc/specs/AccessControl/AccessControlSpecs01.pdf`, accessed on 23 March, 2017.

[73] W. Stallings, "Cryptography and network security: principles and practice", Sixth Edition, Prentice Hall, 2013.

[74] H. M. Sun, W. H. Chang, S. Y. Chang, and Y. H. Lin, "DepenDNS: dependable mechanism against DNS cache poisoning", *Cryptology and Network Security*, Springer, 2009, pp. 174-188.

[75] L. Tawalbeh, M. Mowafi, and W. Aljoby, "Use of elliptic curve cryptography for multimedia encryption", *IET Information Security*, vol. 7, no. 2, 2013, pp. 67-74.

[76]] C. H. Tseng, S. Wang, and W. Tsaur, "Hierarchical and dynamic elliptic curve cryptosystem based self-certified public key scheme for medical data protection", *IEEE Transactions on Reliability*, vol. 64, no. 3, 2015, pp. 1078-1085.

[77] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe, "A survey of mobility in information-centric networks: challenges and research directions", *in Proc. NoM12*, NY, USA, June 2012, pp. 1-6.

[78] M. Vahlenkamp, M. Whlisch, and T. C. Schmidt, "Backscatter from the data plane - threats to stability and security in information-centric networking", *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 57, no. 16, November 2013, pp. 3192-3206.

[79] B. Vieira and E. Poll, "A security protocol for information-centric networking in smart grids", *SEGS13*, ACM, Berlin, Germany, November 2013, pp. 1-10.

[80] Y. Wang, M. Xu, Z. Feng, Q. Li, and Q. Li, "Session-based access control in information-centric networks: design and analyses", *IEEE Performance Computing and Communications Conf. (IPCCC)*, Austin, TX, Dec. 2014, pp. 1-8.

[81] W. Wong and P. Nikander, "Secure naming in information-centric networks", *ACM ReArch*, Philadelphia, USA, 2010.

[82] C. A. Wood, and E. Uzun, "Flexible end-to-end content security in CCN", *IEEE 11th Conference on Consumer Communications and Networking (IEEE CCNC)*, Las Vegas, NV, Jan. 2014, pp. 858-865.

[83] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing", *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, 2013, pp. 843-859.

[84] M. Xie, I. Widjaja, and H. Wang, "Enhancing cache robustness for content-centric networking", *in Proc. of the IEEE Infocom*, Orlando, FL, March 2012, pp. 2426-2434.

[85] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research", *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, 2014, pp. 1024-1049.

[86] X. Yang, D. Wetherall, and T. Anderson, "A DoS-limiting network architecture", *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 35, no. 4, 2005, pp. 241-252.

[87] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane", *Journal of Computer Communications*, vol. 36, no. 7, April 2013, pp. 779-791.

[88] Y. You, M. Zulkernine, and A. Haque, "A distributed defense framework for flooding-based DDoS attacks", *in Proc. of the International Conference on Availability, Reliability and Security*, IEEE CS Press, Barcelona, Spain, 2008, pp. 245-252.

[89] Y. You, M. Zulkernine, and A. Haque, "Detecting flooding-based DDoS attacks", *in Proc. of the IEEE Symposium on Computer and Communications Network Security ICC '07*, Glasgow, Scotland, June 2007, pp. 1239-1234.

[90] J. Yuan and K. Mills, "Monitoring the macroscopic effect of DDoS flooding attacks", *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 4, November 2005, pp. 1–12.

[91] S. T. Zargar, J. Joshi, and D. Tipper, "A Survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks", *IEEE Communications Surveys & Tutorials*, Accepted for publication.

[92] X. Zhang, K. Chang, H. Xiong, Y. Wen, G. Shi, and G. Wang, "Towards name-based trust and security for content-centric network", *19th IEEE International Conference on Network Potocols*, Vancouver, BC, 2011, pp. 1-6.

[93] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: a survey", *Computer Networks*, vol. 57, no. 16, 2013, pp. 3128-3141.

[94] J. Zhang and M. Zulkernine, "Anomaly based network intrusion detection with unsupervised outlier detection", *Symposium on Network Security and Information Assurance in IEEE International Conference on Communications (ICC)*, Istanbul, Turkey, June 2006, pp. 2388-2393.

[95] Anonymizer. [Online]. Available: `http://www.anonymizer.com`, accessed on 23 March, 2017.

[96] Cisco visual networking index: forecast and methodology, 2015-2020, June 2016.

[97] Freedom system 2.0 architecture. [Online]. Available: `http://osiris.978.org/~brianr/crypto-research/anon/www.freedom.net/products/whitepapers/Freedom_System_2_Architecture.pdf/`, accessed on 23 March, 2017.

[98] Future internet assembly, [Online]. Available: `http://www.future-internet.eu/home/future-internet-assembly.html`, accessed on 23 March, 2017.

[99] OWASP risk rating methodology, [Online]. Available: `https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology`, accessed on 23 March, 2017.

[100] Symantec security response – assessing the severity of threats, events, vulnerabilities, security Risks, February 2006, [Online]. Available: `http://www.symantec.com/about/news/resources/press_kits/`

securityintelligence/media/SSR-Severity-Assesment.pdf, accessed on 23 March, 2017.

[101] The network of information: architecture and applications, FP7-ICT-2009-5-257448-SAIL/D-3.1, July 2011, [Online]. Available: http://docplayer.net/12120221-D-3-1-d-b-1-the-network-of-information-architecture-and-applications.html, accessed on 23 March, 2017.

[102] Traces available in the Internet traffic archive, [Online]. Available: http://ita.ee.lbl.gov/html/traces.html, accessed on 23 March, 2017.