Universitat Autònoma de Barcelona

Contributions to Privacy and Anonymity on the Internet Domain Name System and Second-Generation Onion Routing

Sergio Castillo-Pérez

ADVERTIMENT. L'accés als continguts d'aquesta tesi queda condicionat a l'acceptació de les condicions d'ús establertes per la següent llicència Creative Commons: http://cat.creativecommons.org/?page_id=184

ADVERTENCIA. El acceso a los contenidos de esta tesis queda condicionado a la aceptación de las condiciones de uso establecidas por la siguiente licencia Creative Commons: http://es.creativecommons.org/blog/licencias/

WARNING. The access to the contents of this doctoral thesis it is limited to the acceptance of the use conditions set by the following Creative Commons license: https://creativecommons.org/licenses/?lang=en

Universitat Autònoma de Barcelona
Departament d'Enginyeria de la Informació i de les Comunicacions
PhD programme on Computer Science

Contributions to Privacy and Anonymity on the Internet

Domain Name System and Second-Generation Onion Routing

SUBMITTED TO UNIVERSITAT AUTÒNOMA DE BARCELONA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

by Sergio Castillo-Pérez
Bellaterra, March 2017

Advisors
Dr. Joaquin García-Alfaro
Dr. Joan Borrell-Viader

© Copyright 2017 by Sergio Castillo-Pérez

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosopy.

Bellaterra, March 2017

Dr. Joaquin Garcia-Alfaro
Dr. Joan Borrell-Vidader
(Advisors)

Proposed comittee:
Ana Rosa Cavalli, Institut Mines-Telecom/Telecom SudParis
Gregorio Martínez Pérez, Universidad de Murcia
Guillermo Navarro Arribas, Universitat Autònoma de Barcelona
Jose Maria Alcaraz Calero, University of the West of Scotland
Sergi Robles Martínez, Universitat Autònoma de Barcelona

A Mar, Judith y Carla...

Abstract

Everyday all our activity in Internet leaves traces of ourselves and of our way of life through the storing of a big amount of personal data, becoming what we call our *digital identity*. Nowadays, the tracking of the user activities correlated with such digital identities has become one of the principal interest of not only private companies, but also governments. The nature of such interest can obey to different motivations: business, politics, surveillance or censorship among others. Moreover, the exponential growth of the available data and the capabilities to process it has lead to a worst situation. In response to these circumstances, the demand of privacy-preserving and anonymous technologies has been increased on the part of concerned users.

In this line, in the sea of data which we call Internet, some of the underlying protocols that are underpinning its operation are detrimental to the safeguarding of the aforementioned privacy properties. The Domain Name System (DNS) is clearly an example of one of such protocols, specially if we consider that almost every activity on the Internet starts with a DNS query. In fact, when DNS was designed in the early eighties it was not intended to guarantee the privacy of people's queries. In that sense, its underlying design is becoming insufficient to face the changes and innovations of today's Internet.

In contrast, during the recent years, the Tor network has become one of the most popular overlay networks for anonymising TCP traffic. Tor is a low-latency anonymity system that can be installed as an end-user application on a wide range of operating systems, allowing to redirect the traffic through a series of anonymising virtual tunnels. In such a way, users can use network services over Internet without compromising their privacy. Also, it is employed as en extremely effective censorship circumvention tool, allowing to its users to connect against blocked resources.

This dissertation is precisely focused on contributing to these two aforementioned topics —the DNS protocol and the Tor network— by studying the related privacy and anonymity problems and reinforcing current solutions. More precisely, our research efforts are centered on (1) The abuse of the DNS protocol performed by *botnets* and how we can detect such malicious purpose, (2) The lack of privacy of the DNS protocol and how we can improve it, and (3) How we can enhance the performance of the Tor network while security is preserved.

Keywords: Privacy, Anonymity, Domain Name System, Fast-Flux Networks, Private Information Retrieval, The Onion Router, Tor Node Selection Algorithms, Key Agreement Protocols.

Resum

Tots els dies la nostra activitat a Internet deixa rastres de nosaltres mateixos i de la nostra forma de vida a través de l'emmagatzemament d'una gran quantitat d'informació personal, esdevenint en el que anomenem la nostra *identitat digital*. Avui en dia, el seguiment de les activitats dels usuaris correlacionat amb tals identitats digitals s'ha convertit en un dels principals interessos de no només companyies privades, sinó també de governs. La natura d'aquest interessos obeeix a diferents motivacions: comercials, polítiques, vigilància o censura entre d'altres. A més a més, el creixement exponencial de les dades disponibles i les capacitats de processar-les ha conduït a una situació encara pitjor. En resposta a aquestes circumstàncies, la demanda de tecnologies que preserven la privacitat i l'anonimat s'ha incrementat per part dels usuaris preocupats.

En aquesta línia, en el mar de dades que anomenem Internet, alguns dels protocols subjacents que suporten el seu funcionament estan en detriment de la preservació de les propietats de privacitat esmentades. El sistema de noms de domini (DNS) és in clar exemple, especialment si considerem que gairebé totes les activitats a Internet comencen per una petició DNS. De fet, quan el sistema DNS va ésser dissenyat als principis dels vuitanta no es va pretendre garantir la privacitat de les peticions realitzades per persones. En aquest sentit, el seu disseny subjacent està esdevenint insuficient per fer front als canvis i innovacions de la Internet d'avui.

Per contra, durant els últims anys, la xarxa Tor s'ha convertit en una de les xarxes superposades més populars per a l'anonimització de tràfic TCP. Tor és un sistema d'anonimat de baixa latència que es pot instal·lar com una aplicació d'usuari final en una àmplia gamma de sistemes operatius, el que permet redirigir el tràfic a través d'una sèrie de túnels d'anonimització virtuals. D'aquesta manera, els usuaris poden utilitzar els serveis de xarxa a través d'Internet sense comprometre la seva privacitat. A més, s'empra com a eina extremadament eficaç per eludir la censura, el que possibilita als seus usuaris connectar-se a recursos bloquejats.

Aquesta tesi doctoral se centra precisament en contribuir en aquests dos temes abans esmentats —el protocol DNS i la xarxa Tor— a través de l'estudi dels problemes de privacitat i anonimat, i reforçant les solucions actuals. De forma més precisa, els nostres esforços de recerca se centren en (1) L'abús del protocol DNS realitzat per part de *botnets* i com podem detectar tals fins maliciosos, (2) La manca de privacitat del protocol DNS i com podem millorar-la, i (3) De quina manera podem incrementar el rendiment de la xarxa Tor mentre que la seguretat es preserva.

Paraules clau: Privacitat, anonimat, sistema de noms de domini, xarxes Fast-Flux, Private Information Retrieval, The Onion Router, algorismes de selecció de nodes de Tor, protocols d'establiment de claus.

Resumen

Todos los días nuestra actividad en Internet deja rastros de nosotros mismos y de nuestra forma de vida a través del almacenamiento de una gran cantidad de información personal, convirtiéndose en lo que llamamos nuestra *identidad digital*. Hoy en día, el seguimiento de las actividades de los usuarios correlacionado con tales identidades digitales se ha convertido en uno de los principales intereses de no sólo compañías privadas, sino también de gobiernos. La naturaleza de este interés obedece a diferentes motivaciones: comerciales, políticas, vigilancia o censura entre otras. Además, el crecimiento exponencial de los datos disponibles y las capacidades de procesarlos ha conducido a una situación aún peor. En respuesta a estas circunstancias, la demanda de tecnologías que preservan la privacidad y el anonimato se ha incrementado por parte de los usuarios preocupados.

En esta línea, en el mar de datos que llamamos Internet, algunos de los protocolos subyacentes que soportan su funcionamiento están en detrimento de la preservación de las propiedades de privacidad mencionadas. El sistema de nombres de dominio (DNS) es un claro ejemplo, especialmente si consideramos que casi todas las actividades en Internet comienzan por una petición DNS. De hecho, cuando el sistema DNS fue diseñado a principios de los ochenta no se pretendió garantizar la privacidad de las peticiones realizadas por personas. En este sentido, su diseño subyacente está siendo insuficiente para hacer frente a los cambios e innovaciones de la Internet de hoy.

Por el contrario, durante los últimos años, la red Tor se ha convertido en una de las redes superpuestas más populares para la anonimización de tráfico TCP. Tor es un sistema de anonimato de baja latencia que se puede instalar como una aplicación de usuario final en una amplia gama de sistemas operativos, lo que permite redirigir el tráfico a través de una serie de túneles de anonimización virtuales. De esta manera, los usuarios pueden utilizar los servicios de red a través de Internet sin comprometer su privacidad. Además, se emplea como herramienta extremadamente eficaz para eludir la censura, lo que posibilita a sus usuarios conectarse a recursos bloqueados.

Esta tesis doctoral se centra precisamente en contribuir en estos dos temas antes mencionados —el protocolo DNS y la red Tor— a través del estudio de los problemas de privacidad y anonimato, y reforzando las soluciones actuales. De forma más precisa, nuestros esfuerzos de investigación se centran en (1) El abuso del protocolo DNS realizado por parte de *botnets* y cómo podemos detectar tales fines maliciosos, (2) La falta de privacidad del protocolo DNS y cómo podemos mejorarla, y (3) De que manera podemos incrementar el rendimiento de la red Tor mientras que la seguridad se preserva.

Palabras clave: Privacidad, anonimato, sistema de nombres de dominio, redes Fast-Flux, Private Information Retrieval, The Onion Router, algoritmos de selección de nodos de Tor, protocolos de establecimiento de claves.

Acknowledgements

This work has been partially funded by the Ministry of Science and Innovation of Spain, through the project TIN2014-55243-P, and by the Catalan Government with the project 2014SGR691.

This dissertation is the culmination of several years of hard work, and I would like to thank the many people who have made invaluable contributions in some way. First and foremost, I would like to express the deepest appreciation to my advisors Joaquín García and Joan Borrell for giving me the opportunity to perform this PhD. I specially gratitude their patience, kindness, insight and encouragements during these years. I can honestly say that without their talent, guidance, and help this dissertation would not have been possible.

Additionally, I would also like to warmly thank my colleagues of the *department of Information and Communications Engineering* (dEIC) —Autonomous University of Barcelona— who have contributed immensely to my personal and professional career. They have been a constantly source of friendship and inspiration, and with whom I have shared many great times as a researcher and professor. In particular, I specially want to mention Guillermo Navarro, Sergi Robles and M. Carmen de Toro. I also gratefully acknowledge the collaboration of Sergi Martínez during his MSc in dEIC, and in regard to our contribution in the context of botnets and fast-flux networks.

I am indebted to the School of Electrical Engineering of the Aalto University, that hosted me for three months during my research stay in Finland. Profuse thanks to Dr. Jörg Ott for his support during such stay, and his suggestions in the field of identity-based cryptography.

Sin lugar a dudas, debo mencionar a la familia, el pilar fundamental y la más preciada joya que uno puede poseer. En especial, mi más sincera gratitud a mis padres, Pedro y Encarna, por su soporte, dedicación y los valores que me han inculcado. No me olvido de mis suegros, Satur y Ceci, porque su ayuda también ha contribuido a hacer esto posible. Una mención muy especial para las pequeñas Judith y Carla, por darle sentido pleno a mi vida, y ser fuente de motivación, felicidad y amor diario. Por último, y no por ello menos importante, mi más profundo agradecimiento a Mar, por su infinita paciencia en los momentos difíciles, comprensión, apoyo constante, sacrificio, y amor incondicional que han hecho posible que esta tesis llegara a buen puerto.

SERGIO CASTILLO-PÉREZ
March 2017

Contents

List of Figures

List of Tables

List of Abbreviations

ARP Address Resolution Protocol.
ASN Autonomous System Number.
BDH Bilinear Diffie-Hellman.
BGP Border Gateway Protocol.
CBE Certificateless-Based Encryption.
CDH Computational Diffie-Hellman.
CDN Content Delivery Network.
CPU Central Process Unit.
DDoS Distributed Denial of Service.
DGA Domain Generation Algorithm.
DHC Diffie-Hellman Chain.
DHCP Dynamic Host Configuration Protocol.
DNS Domain Name System.
DNSSEC Domain Name System Security Extensions.
DS Directory Server.
DSA Digital Signing Algorithm.
ECC Elliptic Curve Cryptosystem.
ENUM tElephone NUmber Mapping.
ETSI European Telecommunications Standards Institute.
EWMA Exponentially Weighted Moving Average.
HIBE Hierarchical Identity Based Encryption.
HTTP Hypertext Transfer Protocol.
I2P Invisible Internet Project.
IBE Identity-Based Encryption.
ICANN Internet Corporation for Assigned Names and Numbers.
ICMP Internet Control Message Protocol.
IETF Internet Engineering Task Force.
IND-CPA Indistinguishability under Chosen Plaintext Attack.
IoT Internet of Things.
IP Internet Protocol.
IRC Internet Relay Chat.
ISP Internet Service Provider.

ITU-T International Telecommunication Union - Telecommunication.
KGC Key Generator Center.
KSK Key Signing Key.
LAN Local Area Network.
NAPTR Naming Authority Pointer.
NIST National Institute of Standards and Technology.
OR Onion Router.
P2P Peer-to-Peer.
PIR Private Information Retrieval.
PKG Public Key Generator.
pmf Probability mass function.
PPT Probabilistic Polynomial-Tme.
RAM Random Access Memory.
RFC Request for Comments.
RR Resource Record.
RRDNS Round Robin DNS.
RRSIG Resource Record Signature.
RSA Rivest-Shamir-Adleman.
SIP Session Initiation Protocol.
SVM Support Vector Machine.
TCP Transmission Control Protocol.
TLD Top Level Domain.
Tor The onion routing.
TTL Time To Live.
UDP User Datagram Protocol.
URI Uniform Resource Identifier.
URL Uniform Resource Locator.
VoIP Voice over IP.
ZSK Zone Signing Key.

1 Introduction

> “ ***Anonymous**: borrowed into English around 1600 from Late Latin anonymus, from Ancient Greek ἀνώνυμος (anōnumos, “without name”), from ἀν- (an, “without”) + ὄνυμα (onuma), Aeolic dialectal form of ὄνομα (onoma, “name”).* ”

Everyday all our activity in Internet leaves traces of ourselves and of our way of life through the storing of a big amount of personal data, becoming what we call our *digital identity*. Nowadays, the tracking of the user activities correlated with such digital identities has become one of the principal interest of not only private companies, but also governments. The nature of such interest can obey to different motivations: business, politics, surveillance or censorship among others. Moreover, the exponential growth of the available data and the capabilities to process it has lead to a worst situation. As a consequence, the rights to freedom of expression and privacy recognised by the Universal Declaration of Human Rights (*cf.* articles 12 and 19) are infringed in the digital world. In response to these circumstances, the demand of privacy-preserving and anonymous technologies has been increased on the part of concerned users. From networks like Freenet [44], Tor [52] or I2P [144] through digital currencies such as BitCoin [103], CloakCoin [141] or AnonCoin [21], to electronic anonymous voting schemes [102], the technologies related to privacy-preserving and anonymity have not received so much attention till the recent times. This dissertation is precisely focused on contributing to these issues by studying the related privacy and anonymity problems —with special emphasis on DNS and Tor— and by reinforcing current solutions.

In the sea of data which we call Internet, some of the underlying protocols that are underpinning its operation are detrimental to the safeguarding of the aforementioned privacy properties. The Domain Name System (DNS) is clearly an example of one of such protocols, specially if we consider that almost every activity on the Internet starts with a DNS query [72]. In fact, when DNS was designed in the early eighties, it was not intended to guarantee the privacy of people's queries. It was simply conceived as a federated database with information that needed to remain publicly accessible. However, this underlying design is becoming insufficient to face the changes and innovations of today's Internet; for instance, the use of the DNS protocol to lead procedures on VoIP services for the translation of traditional telephone numbers into Internet URLs [72], and the use of the DNS for the resolution of information linked to items of value. Analyses of critical threats to these services can be found in [116, 117].

Threats and vulnerabilities reported in the related literature are indeed an heritage of the vulnerabilities existing in the DNS mechanisms. We can find in [12] and [22] a complete analysis of threats to DNS technologies. The most important threats to DNS technologies can be grouped as follows: (1) authenticity and integrity threats to the trustworthy communication between resolvers and servers; (2) availability threats by means of already existing denial of service attacks; and (3) escalation of privilege due to software vulnerabilities in server implementations. In addition, the DNS protocol uses clear text operations, which means that either a passive attack, such as eavesdropping, or an active attack, such as Man-in-the-Middle, can be carried out by unauthorised users to capture queries and responses. The use of the security extension DNSSEC for DNS, proposed by the IETF in the late nineties, only addresses authentication and integrity problems in the DNS. Although it must certainly be seen as an important asset to enhance the security of DNS applications, it requires to be combined with additional measures to cope the kind of issues discussed previously.

Although there is intensive research work on privacy issues in the Internet community, only few approaches seem to deal with the DNS privacy case scenario. Indeed, beyond limiting and granting access to store people's information, no specific mechanisms have been yet proposed by the Internet community to preserve the invasion of privacy that future lookup services may expose. Seen in this light, the issue of querying a DNS server preserving the privacy of the users can be conceived as a Private Information Retrieval (PIR) problem. The concept of PIR —introduced by Chor, Goldreich, Kushilevitz and Sudan [42, 157]— can be summarised as the problem of retrieving an item from a database without revealing what information is wanted. Therefore, PIR techniques should be identified as a prominent way to protect the lack of privacy of the DNS protocol.

Likewise, the use of anonymity-based infrastructures like the Tor network [52] is often seen as a silver bullet solution to mitigate privacy problems on the Internet. Tor consists of a network of thousands of nodes (or *onion routers*) managed by volunteers that redirect the traffic of low-latency services with a very acceptable overhead. Its implementation is distributed as free software that can be installed as an end-user application on a wide range of operating systems. The objective of Tor is the protection of the privacy of a sender as well as the contents of its messages. To do so, the messages are wrapped in several layers of encryption and sent through a sequence of nodes. Upon reception, each node peels off a layer of encryption and sends the resulting value to the next node. This process is repeated until the last node of the sequence, which recovers the original message, and delivers to the destination. In such a way, each router only knows its adjacent nodes, and any entity that can not view the entire network

is unable to associate the sender of the traffic and the final destination. Despite its popularity, it is well known that the Tor has some design weaknesses as previous research has pointed out. In [5], AlSabah and Goldberg survey those weaknesses and classify previous proposals that aim to address them. In particular, they consider five main research directions to address such design flaws: *congestion*, *router selection*, *scalability*, *circuit construction* and *security*.

Besides the privacy problems described previously regarding the DNS protocol, its infrastructure and lack of security mechanisms can be also abused to render malicious activities. This is the case of the fast-flux networks [73, 154]. Fast-flux networks are a special type of DNS technique used by cyber-criminals to difficult the identification (*i.e.*, the IP address), and to frustrate location and shutting down of servers used for illegal activities (*phishing*, *malware*, *exploit kits*, ...). To achieve this, the domain name registration and name resolution services of the DNS protocol, along with the authenticity threats that we have pointed out previously, are exploited. In particular, each fast-flux network has a fully qualified domain name (also known as *flux domain*) associated to multiple IP addresses that are constantly changing by the modification of the DNS records. This is also accomplished by means of registering the domain with a short TTL (Time-To-Live). Behind each IP associated to the flux domain there is a compromised computer (or *flux agent*) that acts as proxy. Thus, a request to the flux domain will go through one flux agent before being forwarded to the backend server. In this manner, fast-flux networks include an abstraction layer that increases anonymity, availability, load balancing and resiliency to takedown. Taking into consideration this evasion strategy, the detection in real time of fast-flux domains is crucial to warn potential victims before they connect to a malicious site. Hence, research in this scope can help to prevent any malicious activity derived from connections against fast-flux networks.

In the research work presented in this dissertation we have established two main goals. The first one is to study the security problems that the DNS presents with special emphasis on fast-flux networks and privacy. The second objective is focused on the Tor network. In this area, we analyse how the router selection algorithms can influence the degree of anonymity and the network latencies. Additionally, the cryptographic protocols to construct a Tor circuit are explored in order to improve them from the point of view of network performance and scalability. These studies are intended to be contributions to the research directions of router selection, circuit construction and scalability identified in [5].

1.1 Contributions

This dissertation covers the work done in part-time from 2008 to 2013, while I also was developing my professional career in the private sector. This work has been updated analysing the research evolution done during these last years and providing a new result presented in Chapter 7. The contributions of this dissertation can be summarised as follows:

- An innovative strategy based on Support Vector Machines for the detection of fast-flux domains in real time.
- An evaluation of Tor as a DNS privacy-preserving technology.
- An analysis of two previous DNS privacy-preserving strategies based on PIR, and the design,

implementation and evaluation of an original proposal.

- The definition of an original formal model for the characterisation of the Tor node selection algorithms from the point of view of the degree of anonymity.
- An algorithm based on latency graphs that improves the trade-off between degree of anonymity and latency compared to previous proposals.
- A deep review of the key agreement protocols used for the construction of Tor circuits.
- A key agreement protocol for Tor circuit construction based on bilinear pairings that improves the performance and scalability compared to previous proposals.

1.2 List of publications

Positive results of our research have been published in national and international conferences and journals:

- J. GARCIA-ALFARO and S. CASTILLO-PÉREZ. "Resolution of anonymous DNS queries (In Spanish)". In: *X Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*. Sept. 2008
- S. CASTILLO-PÉREZ and J. GARCIA-ALFARO. *Anonymous Resolution of DNS Queries*. In: *On the Move to Meaningful Internet Systems: OTM 2008*. Ed. by R. MEERSMAN and Z. TARI. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, 987–1000. DOI: 10.1007/978-3-540-88873-4_5
- S. CASTILLO-PÉREZ and J. GARCIA-ALFARO. Evaluation of Two Privacy-Preserving Protocols for the DNS. in: *2009 6th International Conference on Information Technology: New Generations*, (2009), 411–416. DOI: 10.1109/ITNG.2009.195
- S. CASTILLO-PÉREZ and J. GARCIA-ALFARO. On the Use of Latency Graphs for the Construction of Tor Circuits. In: *CoRR*, **abs/1208.3730**: (Aug. 2012)
- S. CASTILLO-PÉREZ and J. GARCIA-ALFARO. Onion Routing Circuit Construction via Latency Graphs. In: *Computers & Security*, **37**: (Sept. 2013), 197–214. DOI: 10.1016/j.cose.2013.03.003
- S. MARTINEZ-BEA, S. CASTILLO-PEREZ, and J. GARCIA-ALFARO. "Real-time malicious fast-flux detection using DNS and bot related features". In: *2013 Eleventh Annual Conference on Privacy, Security and Trust*. Institute of Electrical and Electronics Engineers (IEEE), July 2013, 369–372. DOI: 10.1109/PST.2013.6596093
- S. CASTILLO-PÉREZ, J. GARCIA-ALFARO, and J. BORRELL-VIADER. A Scalable and Single-Pass Authenticated Key Agreement Protocol for the Establishment of Second-Generation Onion Routing Circuits. In: *To be submitted*, (2017)

1.3 Document layout

This dissertation can be logically structured in two main parts in correspondence with the research goals that we have postulated. The first one is composed by Chapters 2, 3, and 4, where security aspects regarding the DNS protocol are discussed and specially focused on fast-flux networks and privacy. The second part is composed by Chapters 5, 6 and 7, and provides some research results related to the improvement of the Tor network from the standpoint of performance and scalability. Below, some more details about the content of each chapter are provided:

- **Chapter 1: Introduction.** This chapter presents the context and the motivation of this dissertation.
- **Chapter 2: Preliminaries.** This chapter includes all the necessary concepts, notations, models, and core definitions that are the cornerstone to understand the rationale of our work.
- **Chapter 3: DNS and fast-flux networks.** This chapter analyses the security of the DNS protocol from the perspective of the fast-flux networks. We also propose a new classifier based on Support Vector Machines that improves the real-time detection of fast-flux domains.
- **Chapter 4: Privacy in DNS with Tor and PIR.** In this chapter the privacy problem of the DNS is studied. In particular, the use of the Tor network is considered and evaluated as a way to protect the privacy and anonymity of the users. Also, two previous proposals based on PIR are analysed, and a new one is presented and evaluated.
- **Chapter 5: Formal modelling of Tor node selection criteria.** Motivated by the need to enhancing the performance of the Tor network, in this chapter the Tor node selection criteria are studied. Such selection criteria are presented as an inherent potential source of performance degradation. We start by proposing a new formal way of modeling the Tor node selection criteria. This formal model allows us to obtain an algebraic way to compute the degree of anonymity provided by the selection algorithms of Tor. In conjunction with the measurements of empirical latencies, the formal model provides an extremely useful tool in order to choose an algorithm based on the trade-off between the degree of anonymity and latency.
- **Chapter 6: Latency graphs for the Tor circuit construction.** This chapter presents a new Tor node selection algorithm based on a concept that we call latency graphs. The analysis of the new algorithm shows that it outperforms other classical strategies from the point view of the trade-off between degree of anonymity and latency. This analysis is sustained on the formal model presented in the previous chapter.
- **Chapter 7: Scalable and single-pass key agreement protocol for Tor circuit establishment.** This chapter tackles the particular issue of the construction of the Tor circuits as another source of performance penalty and lack of scalability. A deep review of the state of the art is presented and a new protocol based on bilinear pairings is proposed. The protocol is also proven secure under a formal model.
- **Chapter 8: Conclusions and open problems.** This chapter summarises the main conclusions of this dissertation and gives an outlook to some future lines of research.

2

Preliminaries

> *We can only see a short distance ahead, but we can see plenty there that needs to be done.*
>
> ALAN TURING

In this chapter we introduce some preliminaries that will be used throughout the rest of this dissertation. The aim of these preliminaries is to provide an overview of the main topics, making this document self contained. The first section covers some mathematical aspects that will be relevant for Chapters 5 and 7. Following, the DNS protocol and its threats are analysed along with the DNSSEC extension. The next section is devoted to the botnets, including their life cycle and architecture designs. Finally, the last part deals with the primordially concepts about the Tor network. The reader can come back to this chapter when reading the following ones, using it as a point of reference. We encourage the readers to consult related literature for further details.

2.1 Mathematical background

This section is intended to be a review of the main mathematical concepts and definitions needed in the following chapters. It should not be understood as a deep description of all the related aspects, but can be used as a basic reference.

2.1.1 Probability spaces

The main purpose of the probability theory is to model random experiments in such a manner that we can obtain inferences about them. For this purpose, a fundamental mathematical object is used in order to describe the experiment or collection of experiments. This object is known as *probability space*. Let us formalise this concept:

Definition 1 (Probability space). *A probabilistic space is defined by a triple* $(\Omega, \mathcal{F}, \mathbb{P})$ *which each element is described as follows:*

- Ω *is a sample space or, in other words, the set of all possible outcomes of the experiment. These elements are usually denoted by* ω*, and called elementary outcomes.*
- $\mathcal{F}$ *is a* σ*-field, a collection of subsets of* Ω*. Sets in* $\mathcal{F}$ *are called events.*
- $\mathbb{P}$ *is a probability measure, a function that assigns a probability to every set in the* σ*-field* $\mathcal{F}$*, with* $\mathbb{P}(\Omega) = 1$*, and such that if* $E_1, E_2, ... \in \mathcal{F}$ *are disjoint events, meaning that* $E_i \cap E_j = \varnothing$ *whenever* $i \neq j$*, then:*

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} \mathbb{P}[E_i]$$

Sometimes it is helpful to consider the simpler case where the sample space Ω is finite or countable. Thus, we define the *discrete probability* space as follows:

Definition 2 (Discrete probability space). *A discrete probability space is a tripe* $(\Omega, \mathcal{F}, \mathbb{P})$ *such that:*

- *The sample space* Ω *is finite or countable, that is* $\Omega = \{\omega_1, \omega_2, \ldots\}$.
- *The* σ*-field* $\mathcal{F}$ *is the set of all subsets of* Ω*.*
- *The probability measure* $\mathbb{P}$ *assigns a number in the set* $[0, 1]$ *to every subset of* Ω*, and defined in terms of the probabilities of the elementary outcomes* $\mathbb{P}(\{\omega\})$*. Also, it is satisfied that:*

$$\mathbb{P}(A) = \sum_{\omega \in A} \mathbb{P}(\{\omega\}),$$

for every $A \subset \Omega$*, and*

$$\sum_{\omega \in \Omega} \mathbb{P}(\{\omega\}) = 1.$$

Associated to the concept of probability spaces, the notion of random variable is also used to make predictions based on data obtained from scientific experiments. Informally, a random variable is a function that maps all possible outcomes of a random experiment into a measurable space (in general, into the real numbers space). In spite of random variables can be defined continuous (*i.e.* they take values within a range), we focus our attention in the discrete version or, in other words, those random variables that their range is finite or countable.

Definition 3 (Discrete random variable). *A discrete random variable* X *on a probability space* $(\Omega, \mathcal{F}, \mathbb{P})$ *is a function* $X : \Omega \to \mathbb{R}$ *such that the range of X is finite or countable and for each* $x \in \mathbb{R}$*,* $\{\omega \in \Omega : X(\omega) = x\} \in \mathcal{F}$*. Also the probability mass function (pmf) of the discrete random variable* X *is defined as:*

$$f(x) = \mathbb{P}(X = x) = \mathbb{P}(\omega \in \Omega : X(\omega) = x)$$

2.1.2 Concepts from abstract algebra

Abstract algebra occupies a central role in cryptography. In this connection, algebraic structures, composed by sets of elements and operations applied to them, constitute a fundamental an extremely powerful tool. We offer to the reader an overview of some classical concepts that would help to the reader to understand our work.

Definition 4 (Binary operation). *A binary operation $*$ on a set $\mathbb{S}$ is a function $f : \mathbb{S} \times \mathbb{S} \to \mathbb{S}$ or, in other words, a function that assigns to each pair of elements $a, b \in \mathbb{S}$ a unique value $a * b \in \mathbb{S}$.*

Definition 5 (Group). *A group $(\mathbb{G}, *)$ is a non-empty set $\mathbb{G}$ and a binary operation $*$ that has the following properties:*

- *Closure: $\forall a, b \in \mathbb{G}$ the element $a * b$ is a uniquely defined element of $\mathbb{G}$.*
- *Associativity: $\forall a, b, c \in \mathbb{G}$, $a * (b * c) = (a * b) * c$.*
- *Identity element: There exists an element $e \in \mathbb{G}$ such that $\forall a \in \mathbb{G}$, $e * a = a$ and $e * a = a$.*
- *Inverse element: For each $a \in \mathbb{G}$ there exists an inverse element $a^{-1} \in \mathbb{G}$ such that $a * a^{-1} = e$ and $a^{-1} * a = e$.*

Definition 6 (Abelian group). *A group $(\mathbb{G}, *)$ with the additional property (commutativity) $a * b = b * a$ for all $a, b \in \mathbb{G}$ is called an Abelian group.*

Definition 7 (Order). *If $(\mathbb{G}, *)$ is a group, then the order (of $\mathbb{G}$) denoted by $|\mathbb{G}|$, is the number of elements in the set $\mathbb{G}$, and which can be either finite or infinite.*

Definition 8 (Cyclic group). *A group $(\mathbb{G}, *)$ is cyclic if there exists an element $\alpha \in \mathbb{G}$ such that for any $b \in \mathbb{G}$ there exists an integer i such that we can write $b = \alpha^i$. Such element α is called a generator of $\mathbb{G}$ and we use the notation $\mathbb{G} = \langle \alpha \rangle$.*

Definition 9 (Field). *A field $(F, +, \times)$ is a set $\mathbb{F}$ and two binary operations $+$ and $\times$ on $\mathbb{F}$ that have the following properties for all $a, b, c \in \mathbb{F}$:*

- *$(\mathbb{F}, +)$ is an Abelian group.*
- *Let $\mathbb{F}^*$ be the set of elements of $\mathbb{F}$ except the identity element for the operation $+$. The $(\mathbb{F}^*, *)$ is an Abelian group.*
- *$\mathbb{F}$ satisfies the distributive law: $a * (b + c) = a * b + a * c$.*

Definition 10 (Finite field). *A finite field is a field that contains a finite number of elements. It was proven by Galois that any field with a finite number of elements has a number of elements equals to q^m for some prime q and some natural number m. There is exactly one finite field for any given size q^m, and we use the notation $\mathbb{F}_{q^m}$.*

Definition 11 (Characteristic and extension degree of a field). *Let $\mathbb{F}_p$ a finite field, if $p = q^m$ where q is a prime and $m \in \mathbb{Z}_n$, then q is called the characteristic of $\mathbb{F}_q$ and m is called the extension degree of $\mathbb{F}_q$.*

Definition 12 (Extension field). *Let $\mathbb{F}_q$ be a finite field with a prime q, the field $\mathbb{F}_{q^m}$ with an integer $m > 1$ is defined as an extension field of the subfield $\mathbb{F}_q$.*

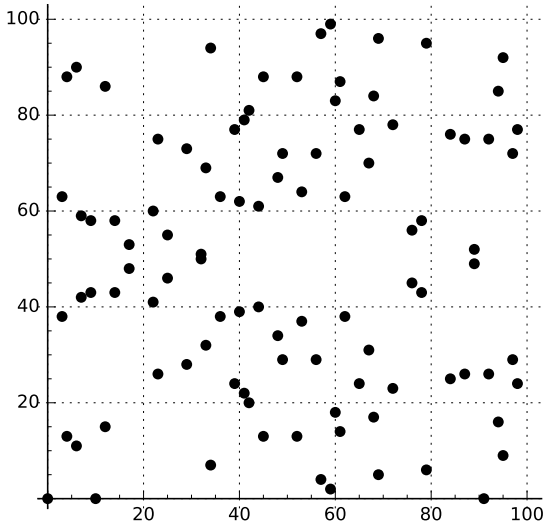


Figure 2.1: Elliptic curve $y^2 = x^3 - x$ over $\mathbb{Z}_{101}$

2.1.3 Elliptic curves

Some of the key agreement protocols that we will study in Chapter 7 are build on pairings of elliptic curve groups. In this section we review some basic concepts regarding elliptic curves and their properties.

Definition 13 (Elliptic curve)**.** *An elliptic curve is defined by the equation of the form:*

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

over the real numbers.

Since the definition over the real numbers is not efficient from the standpoint of the cryptography, we can limit the curve to elements of finite fields. This allow us to generate curves that only operate with integer points. As an example, in Figure 2.1 we show the definition of the elliptic curve $y^2 = x^3 - x$ over the field $\mathbb{Z}_{101}$. In this way, we can introduce the next definition:

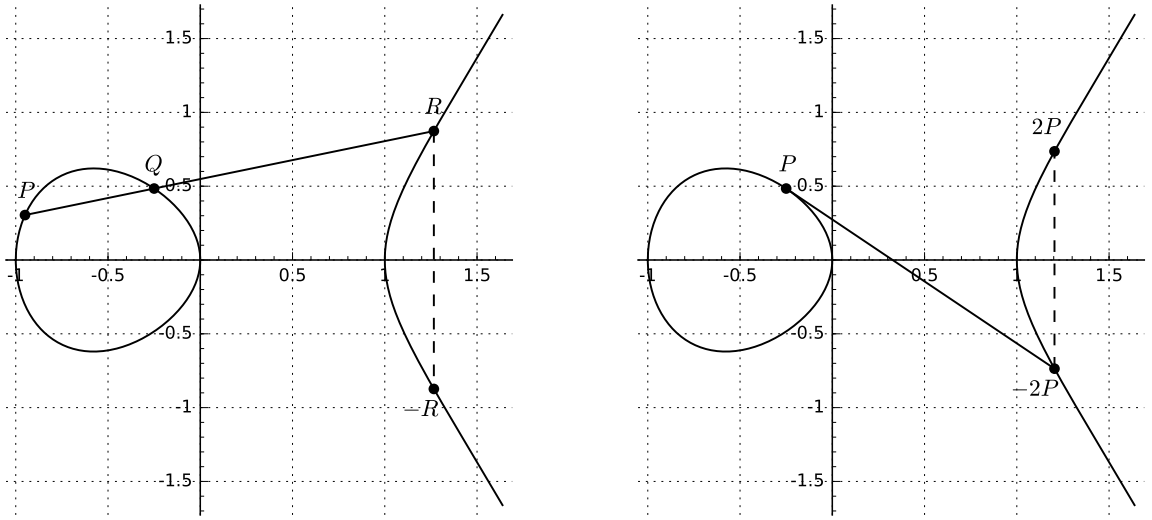
Definition 14 (Elliptic curve over $\mathbb{F}_p$)**.** *Let $\mathbb{F}_p$ be a finite field with p an odd prime number, and let $a, b \in \mathbb{F}_p$ holding the condition $4a^3 + 27b^2 \not\equiv 0$ (mod p). Then, an elliptic curve over $\mathbb{F}_p$ given $a, b \in \mathbb{F}_p$, consists of the points $P = (x, y)$ for $x, y \in \mathbb{F}_p$ that satisfy the equation:*

$$y^2 \equiv x^3 + ax + b \text{ (mod } p\text{)}$$

and we use the notation $E(\mathbb{F}_p)$.

Given the concept of elliptic curve over a finite field, we can construct an algebraic structure of a group, which has some interesting properties from the perspective of security. Hence, we introduce the following definition:

Definition 15 (Elliptic curve over an extension field)**.** *Given an elliptic curve E and an extension field $\mathbb{K}$ of $\mathbb{F}_p$ (p an odd prime) together with a point $\mathcal{O}$ called the point at infinity, the set $\{(x, y) \in \mathbb{K}\times\mathbb{K} : E(\mathbb{K})\}\cup\{\mathcal{O}\}$ has an structure of an algebraic group under some defined group operation.*



(a) Point addition **(b)** Point doubling

Figure 2.2: Geometric interpretation of point addition and point doubling operations in elliptic curves

In order to construct an elliptic curve group $E(\mathbb{K})$ we will use the group operation called *point addition*, which is defined geometrically, and that provides the following properties:

- Let $P = (x, y)$ and $Q = (x', y')$, where $P, Q \in E(\mathbb{K})$. The *point addition* of P and Q, denoted $-R$ is defined as follows (*cf.* Figure 2.2a). Let a line that pass through the points P and Q and intersects the curve in the a third point R, then $P + Q$ is the reflection of this point in the x-axis.
- $P + \mathcal{O} = \mathcal{O} = P$ for all $P \in E(\mathbb{K}_p)$. This $\mathcal{O}$ is the additive *identity* of the group.
- Let the points $P = (x, y)$ and $Q = (x, -y)$. Then $Q = -P$ and $P + Q = P - P = \mathcal{O}$. Then the *inverse* of P is $-P$.
- Let $P = (x, y)$ and $Q = (x', y')$, if $x = x'$ but $y \neq y'$, then $P + Q = \mathcal{O}$.
- Let $P = (x, y)$, then the *point doubling* of P is defined geometrically as follows (*cf.* Figure 2.2b). Consider the tangent of P and the point R where it intersects with the curve. The double of P is the reflection point of R in the x-axis.
- $E(\mathbb{F}_p)$ is *commutative* since $(P + Q) + R = P + (Q + R)$ and *associative* because $P + Q = Q + P$.

Additionally, we can define a *scalar multiplication* operation of elliptic curve points. Given an integer a and a point $P \in E(\mathbb{F}_p)$, the scalar multiplication —denoted by aP— is the process of adding P to itself a times, *i.e.* $P + ... + P = aP$. This operation can be computed efficiently using the addition rule with the double-and-add algorithm. Also, it is believed that is infeasible to reverse the operation.

2.1.4 Bilinear maps

Nowadays, bilinear maps have become one important primitive in security, allowing to build new cryptosystems and protocols such as the identity-based encryption or the tripartite Diffie-Hellman scheme.

In spite of that we define the bilinear map in a general sense, it is important to remark that it is possible to build them over elliptic curves.

Definition 16 (Bilinear map)**.** *Let us consider two additive cyclic groups* $(\mathbb{G}_1, +)$ *and* $(\mathbb{G}_2, +)$*, and a multiplicative cyclic group* $(\mathbb{G}_T, *)$*, all of them with the same prime order* n*. A bilinear map is a map* $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ *with the following properties:*

- *Bilinearity: For all* $P \in \mathbb{G}_1$*,* $Q \in \mathbb{G}_2$ *and* $a, b \in \mathbb{Z}_n$*, we have the relation* $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$*.*
- *Non-degeneracy: The map does not send all pairs in* $\mathbb{G}_1 \times \mathbb{G}_2$ *to the unity in* $\mathbb{G}_T$*.*
- *Computability: There is an efficient algorithm to compute* $\hat{e}(P, Q)$ *for any* $P \in \mathbb{G}_1$ *and* $Q \in \mathbb{G}_2$*.*

There is a special form of bilinear parings called *symmetric pairings*. Such pairings are characterised by the property $\mathbb{G}_1 = \mathbb{G}_2$, and thus $\hat{e}(P, Q) = \hat{e}(Q, P)$ for any $P, Q \in \mathbb{G}_1$. The cryptographic protocols that we will present in this dissertation use symmetric bilinear pairings and, for the sake of simplicity, we denote $\mathbb{G}_1 = \mathbb{G}_2$ by $\mathbb{G}$.

2.1.5 Computational problems

We introduce in this section the computational problems that will form the fundamental pillars of many of the security schemes and protocols covered in this dissertation, and that are mainly based on number theoretic problems. Such problems and their associated assumptions are encompassed within the framework of complexity theory. In that sense, we define the terms of *PPT adversary* and *negligible function*. Both terms permit to define security against a specific set of adversaries whose computational power is bounded. In general terms, it is assumed an adversary that is bounded to run an attack in time polynomial to k, where k is defined as the security parameter. In such a way, we are able to to express the notion of breaking security probabilistically and in terms of a security parameter. The following definitions and concepts will help to understand better such ideas.

Definition 17 (Polynomial-time algorithm)**.** *An algorithm running on an input of length* k *is said be polynomial-time if its running time is* $f(k) = \mathcal{O}(k^c)$*. This is equivalent to say that for some positive constant* c *and for some* a*, we have that* $f(k) < ak^c$ *for all* $k > k_0$*.*

Associated to the concept of polynomial-time algorithm we have the notion of PPT algorithm:

Definition 18 (Probabilistic Polynomial-Time algorithm (informal definition))**.** *A Polynomial-Time algorithm that is randomised, i.e. it employs a certain degree of randomness as part of its logic, is referred as a Probabilistic Polynomial-Time (PPT) algorithm.*

Definition 19 (Negligible function)**.** *A function* $\eta(\cdot)$ *is called negligible if for all* $c > 0$ *there exists a* k_0 *such that* $\eta(k) < 1/k^c$ *for all* $k > k_0$*.*

Given the concept of bilinear pairings, PPT adversaries, and negligible function, we proceed with the definition of a series of computational problems and assumptions that are related with our research. They are specially relevant for the topics covered in Chapter 7.

Definition 20 (The Discrete Logarithm problem)**.** *Let $\mathbb{G}$ be a group of primer order n and P a generator of $\mathbb{G}$. The Discrete Logarithm Problem (DLP) is as follows: given aP for a uniform $a \in \mathbb{Z}_n^*$ compute a.*

Definition 21 (The Discrete Logarithm assumption)**.** *An algorithm $\mathcal{A}$ is said that has advantage $\epsilon(k)$ in solving the DLP problem for $\langle n, \mathbb{G} \rangle$ if the following expression holds:*

$$Pr[\mathcal{A}(k, \mathbb{G}, aP) = a] \geq \epsilon(k)$$

where $k \in \mathbb{N}$ is the bit-length of n and called the security parameter. If for every polynomial-time (in k) algorithm to solve the DLP problem on $\langle n, \mathbb{G} \rangle$, the advantage $\epsilon(k)$ is a negligible function, then $\langle n, \mathbb{G} \rangle$ is said to satisfy the DLP assumption.

Definition 22 (The Computational Diffie-Hellman problem)**.** *Let $\mathbb{G}$ be a group of primer order n and P a generator of $\mathbb{G}$. The Computational Diffie-Hellman Problem (CDH) is as follows: given $\langle P, aP, bP \rangle$ for some $a, b \in \mathbb{Z}_n^*$ compute $abP \in \mathbb{G}$.*

Definition 23 (The Computational Diffie-Hellman assumption)**.** *An algorithm $\mathcal{A}$ is said that has advantage $\epsilon(k)$ in solving the CDH problem for $\langle n, \mathbb{G} \rangle$ if the following expression holds:*

$$Pr[\mathcal{A}(k, \mathbb{G}, aP, bP) = abP] \geq \epsilon(k)$$

where $k \in \mathbb{N}$ is the bit-length of n and called the security parameter. If for every polynomial-time (in k) algorithm to solve the CDH problem on $\langle n, \mathbb{G} \rangle$, the advantage $\epsilon(k)$ is a negligible function, then $\langle n, \mathbb{G} \rangle$ is said to satisfy the CDH assumption.

Definition 24 (The Bilinear Diffie-Hellman problem)**.** *Let $\mathbb{G}$ and $\mathbb{G}_T$ two groups of primer order n. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ a bilinear map and let P be a generator of $\mathbb{G}$. The Bilinear Diffie-Hellman problem (BDH) [78, 24] is as follows: given $\langle P, aP, bP, cP \rangle$ for some $a, b, c \in \mathbb{Z}_n^*$ compute $\hat{e}(P, P)^{abc} \in \mathbb{G}_T$.*

Definition 25 (The Bilinear Diffie-Hellman assumption)**.** *An algorithm $\mathcal{A}$ is said that has advantage $\epsilon(k)$ in solving the BDH problem for $\langle n, \mathbb{G}, \mathbb{G}_T, \hat{e} \rangle$ if the following expression holds:*

$$Pr[\mathcal{A}(k, \mathbb{G}, \mathbb{G}_T, aP, bP, cP) = \hat{e}(P, P)^{abc}] \geq \epsilon(k)$$

where $k \in \mathbb{N}$ is the bit-length of n and called the security parameter. If for every polynomial-time (in k) algorithm to solve the BDH problem on $\langle n, \mathbb{G}, \mathbb{G}_T, \hat{e} \rangle$, the advantage $\epsilon(k)$ is a negligible function, then $\langle n, \mathbb{G}, \mathbb{G}_T, \hat{e} \rangle$ is said to satisfy the BDH assumption.

2.2 DNS and its threats

Nowadays, the DNS infrastructure has become one of the most important pillars in the context of networks, specially if we consider that almost every connection is preceded by the use of its underlying protocol. Nevertheless, it suffers from certain lack of security. In fact, when the DNS protocol was designed, it was not intended to guarantee privacy to people's queries. This makes sense if we consider that DNS is conceived as a distributed hierarchical database which information must be accessed publicly. In scenarios where the DNS protocol is used for the mapping of host and domain names towards traditional Internet services, the inference of information by observing queries and responses can fairly

be seen as acceptable — from the point of view of people's privacy. Nevertheless, the use of the DNS protocol on new lookup services, such as the ENUM suite of protocols, clearly introduces a new dimension. Vulnerabilities on the DNS, allowing the disclosure of data associated with people's information, such as their telephone numbers, is a critical threat [116, 117]. Let us summarise these privacy weaknesses from the following three different scopes: (1) DNS local resolvers, (2) communication channel, and (3) remote DNS servers.

On the first hand, Zhao *et al.* identify in [160] some privacy threats related with local malware targeting the client. Applications such as keyloggers, trojans, rootkits and so on can be considered as a way to obtain the relation between DNS queries and the client who launches them. Let us note that our work does not address the specific case of malware targeting the privacy of the DNS service at the client side. On the second hand, we can identify two main threats targeting the communication channel: (1) passive eavesdropping and (2) active attacks against the network traffic. In the first case, the eavesdropping of plaintext DNS traffic flowing across unprotected wired or wireless LAN networks can be used as a form of anonymity violation. In the second case, traffic injection can also be used to attack the privacy. These attacks can be used to redirect the traffic to a malicious computer, such as ARP spoofing, ICMP redirect, DHCP spoofing, port stealing, etc. Thus, an attacker can redirect every query to a malicious DNS server with the objective of impersonating the correct one and, as a result, to compromise the client privacy. On the third hand, the existence of dishonest or malicious servers can also reduce the level of privacy. Indeed, the DNS cache model allows intermediate servers to maintain a query-response association during a given period of time. The expiration time of every entry in the cache of a server is based on the IP TTL field of a DNS response — as it is defined in [98]. During this period of time, if a client queries a cached entry, the response will be served without any additional resolution. Otherwise, after this time has elapsed, the entry is removed from the cache and, if a client requests it again, the server resolves it, caches it, and sends the response to the client.

Under certain conditions, the observation of the TTL field can be used by attackers to infer the relation between a client and a particular query, reducing the level of anonymity. If attackers suspect that a given client has launched a specific query, they can resolve the same query on the server used by the client. After the response has been retrieved by the attackers, they can determine the current cache expiration time provided by the server. If the returned value is the maximum expiration time defined by the authoritative server, the attackers can deduce that the query has not been launched by the client in, at least, a period that equals the maximum cache expiration time. However, if the value is less than the TTL value, the attackers can consider, with a certain level of probability, that this query was made by the client at most at *maximum expiration time* minus *current expiration time*. This strategy can be applied by potential attackers under certain circumstances. First of all, it can only be considered in networks composed by a few number of clients and/or a DNS server that receives few queries by these clients. Otherwise, the probability of a correct correlation between the specific query and a given client must be considered almost zero. Secondly, if the expiration time defined by the authoritative server has a low value, it can lead to a situation where attackers might launch the query after it expires in the DNS cache (previously created by the client).

2.2.1 The DNSSEC specifications

The Domain Name System SECurity (DNSSEC) extension is a set of specifications of the IETF for guaranteeing authenticity and integrity of DNS Resource Records (RRs) such as NAPTR records. DNSSEC is based on the use of asymmetric cryptography and digital signatures. DNSSEC is often criticised for not being yet deployed after several years of discussions and revisions. It is however the best available solution (when used properly) to mitigate active attacks against the DNS, such as Man-in-the-Middle and cache poisoning. DNSSEC only addresses threats on the authenticity and integrity of the service. Although early DNSSEC proposals presented clear problems of management associated with its key handling schema, the latest established version of DNSSEC overcomes key management issues based on the Delegation Signer (DS) model proposed in RFCs 3658 and 3755.

The main characteristics of the latest version of DNSSEC are described in RFCs 3658, 3755, 4033, 4034, and 4035. An analysis of threats addressed and handled by DNSSEC is also available in RFC 3833. DNSSEC provides to DNS resolvers origin authentication of Resource Records (RRs) (such as A, CNAME, MX, and NAPTR), as well as RR integrity and authenticated denial of existence (*e.g.*, if a NAPTR record is queried in the global DNS service and it does not exist, a signed proof of non-existence is returned to the resolver). As we pointed out above, DNSSEC allows two different strategies to guarantee authenticity and integrity. On the one hand, administrators of a given domain zone can digitally sign their zones by employing their own private key and making available to resolvers the corresponding public key. On the other hand, administrators can rely on the use of a chain of trust between parent and child zones that enables resolvers to verify when the responses received from a given query are trustworthy. In order to implement these two strategies, DNSSEC relies on the use of four new DNS RR types: (1) Resource Record Signature (RRSIG) RRs that store the signature associated to every RR in a given zone, (2) DNS Public Key (DNSKEY) RR that contains the specific public key that will allow the resolver to validate the digital signatures of each RR, (3) Delegation Signer (DS) RRs that are added in parent zones to allow delegation functions on child zones, and (4) Next Secure (NSEC) RRs that contain information about the next record in the zone, and that allow the mechanism for verifying the non-existence of RRs on a given zone. DNSSEC includes two bit flags unused on DNS message's headers to indicate (1) that the resolver accepts unauthenticated data from the server and (2) that those RRs included in the response were previously authenticated by the server.

Regarding the set of keys for signing RRs, one or two key pairs must be generated. If administrators decide to sign zones without a chain of trust, the complete set of RRs of each zone are signed by using a single pair of Zone Signing Keys (ZSKs). On the other hand, if the administrators decide to use a chain of trust between parent and child zones, two key pairs must be generated: a pair of Key Signing Keys (KSKs) is generated to sign the top level DNSKEY RRs of each zone; and a pair of ZSKs keys are used to sign all the RRs of each zone. Several algorithms can be used for the generation of key pairs, such as RSA, DSA (Digital Signature Algorithm), and ECC (Elliptic Curve Cryptosystem). These keys are only used for signatures, and not for encryption of the information. The type and length of these keys must be chosen carefully since it significantly affects the size of the response packets as well as the computational load on the server and the response latency.

The validity period associated with KSK/ZSK keys must also be defined carefully in order to avoid problems with key rollovers, since data signed with previous keys may still be alive in intermediary

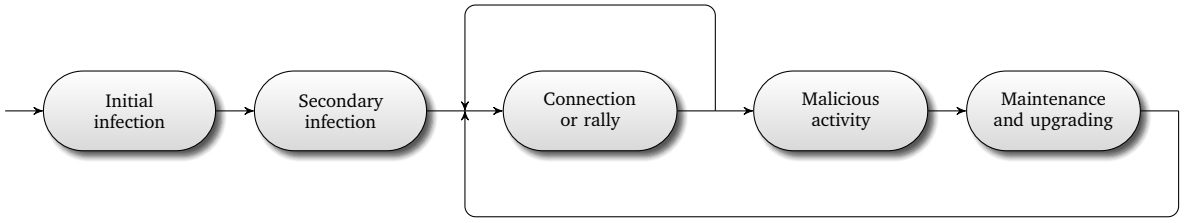


Figure 2.3: Phases of the life cycle of a bot

caches. Synchronisation parameters are therefore very important in DNSSEC. Another issue, often referred in the literature as *zone enumeration* or *zone walking*, relies on the use of the NSEC RR. As we pointed out above, NSEC allows chaining the complete set of RRs of a zone to guarantee non-existence of records and so, it also allows retrieving all the information associated to a given zone. Although the DNSSEC working group originally stated that this is not a real problem (since, by definition, DNS data is or should be public) they proposed an alternative method that uses a new RR called NSEC3 which prevents trivial zone enumeration to introduce a signed hash of the following record instead of including directly its name. Secure storage of trust anchors has also been actively discussed in the literature. Unlike PKI solutions, the chain of trust of DNSSEC offers higher benefits compared to the security of X.509 certificates since the number of keys to protect in DNSSEC is much lower.

2.3 Botnets

Botnets are defined as networks constituted by slave computers —also know as *bots*, *zombies* or *agents*— that have been infected with a malicious software. Most of these infected computers are home-based systems connected to Internet for long periods of time, and without robust protection mechanisms. The executed software allows to an operator (also called *botmaster* or *botherders*) to remotely control the compromised systems, and to perform several illegal activities by using their associated resources (network bandwidth, computation, ...). In order to perform this, a critical component called *Command and Control* (C&C) infrastructure is used. This infrastructure allows to the botmaster to control and coordinate the zombies. Also, it allows to the bots to return the results of their actions to the botmaster. In the recent years, botnets have become a serious threat to the Internet and its users, specially if we consider that behind them there is an extremely lucrative business model for cyber-criminals.

2.3.1 Life cycle of zombies

According to Silva *et al.* [130], a clean system must go through a cycle of phases before it becomes an active bot. These phases are depicted in Figure 2.3, and is comprised by five states: *initial infection*, *secondary injection*, *connection or rally*, *malicious activities* and *maintenance and upgrading*.

The first phase, named *initial infection*, is performed when a clean host is infected with a malicious software (*i.e. malware*) that can lead the system to become a zombie. The vectors of infection are the same of any other *malware*, which comprises, among others, the use of social engineering and attached files in email messages, websites that exploit vulnerabilities on some components of the web browsers (also known as *exploit kits*), or infected removable storage devices.

The second phase, or the *secondary injection*, is performed if the first one has been completed successfully. In this step, the malware executed by the infected host tries to download a new malicious binary. When it is downloaded and executed, this code makes the host to become a new zombie of the botnet. To perform the download of such new binary, the repository servers can be contacted by different ways. In this aim, the malicious software executed can contain an encoded static list of IP addresses, or as list of domain names which can be static or dynamic. It is clear that the dynamism related to the use of domains names gives a greater strategy to the botnet, since it increases the difficult to take down or block this phase. It is worth mentioning that if domain names are used, the execution of the DNS protocol is required. Therefore, the monitoring of the DNS traffic can be used as a mechanisms to identify the botnet. Since the previous phase and this one are essentially related, sometimes are executed simultaneously and grouped in a single step.

The third phase, referred as *connection or rally*, is described as the process of establishing a connection with the C&C servers. The goal is to ensure that the botmaster knows that this particular bot is taking part of the botnet, and that is available to receive command with the aim of performing malicious tasks. This phase is scheduled for its execution every certain period of time, or when the system that lodges the malware is restarted. This phase is considered critical, since it is possible to use strategies that identify network traffic patterns that reveal the elements of the botnet.

During the *malicious activities* phase the botmaster sends —using the C&C channel— commands to the zombies in order to perpetrate illegal activities. There is a wide range of possibilities: performing Distributed Denial of Services (DDoS) attacks, spreading SPAM, stealing of personal or financial information, or bitcoin mining. The botnet can be also used to distribute malware, including the one that can be used to turn a system into a zombie and make the botnet grow. At this point, the exchanged messages can be increased by means of the C&C infrastructure. Notwithstanding, the volume of traffic is not excessively high which, according to some research [60], does not allow to use anomaly-based techniques to identify the infected systems.

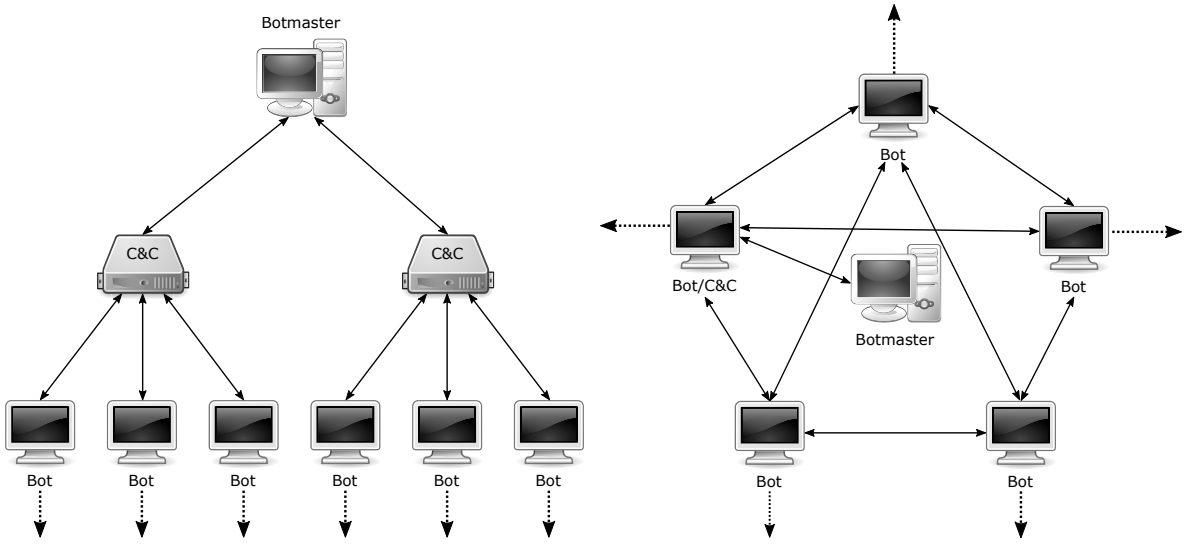
The last phase, known as *maintenance and upgrading*, aims to preserve the bots alive by updating the malicious code that they execute. This permits to the botmaster to integrate new features, avoid its detection by evading anti-malware software, or migrate to another C&C. This phase is usually considered as critical, since the upgrade process can reveal some evidences that makes the botnet detectable.

2.3.2 Architectural designs

Depending on how the command and control infrastructure is designed, the botnets can be classified in accordance to the following architectures (*cf.* Figures 2.4a, 2.4b, 2.5a and 2.5b):

- *Centralised C&C*: In this architecture the bots establish the communication with just only one, or few, servers. Since the connection is performed against a particular server (or servers), this incurs in a central point of failure, which increase the facility to detect and take down the botnet. Typical examples of this type of scheme are those implemented through the use of the Internet Relay Chat (IRC).

- *Decentralised C&C*: In order to increase the flexibility and robustness, this approach can be adopted by botnets. It relies on a variety of P2P (Peer-to-Peer) protocols and working as an overlay network.



(a) Centralised C&C **(b)** Decentralised C&C

Figure 2.4: Architectures of the botnets (1/2)

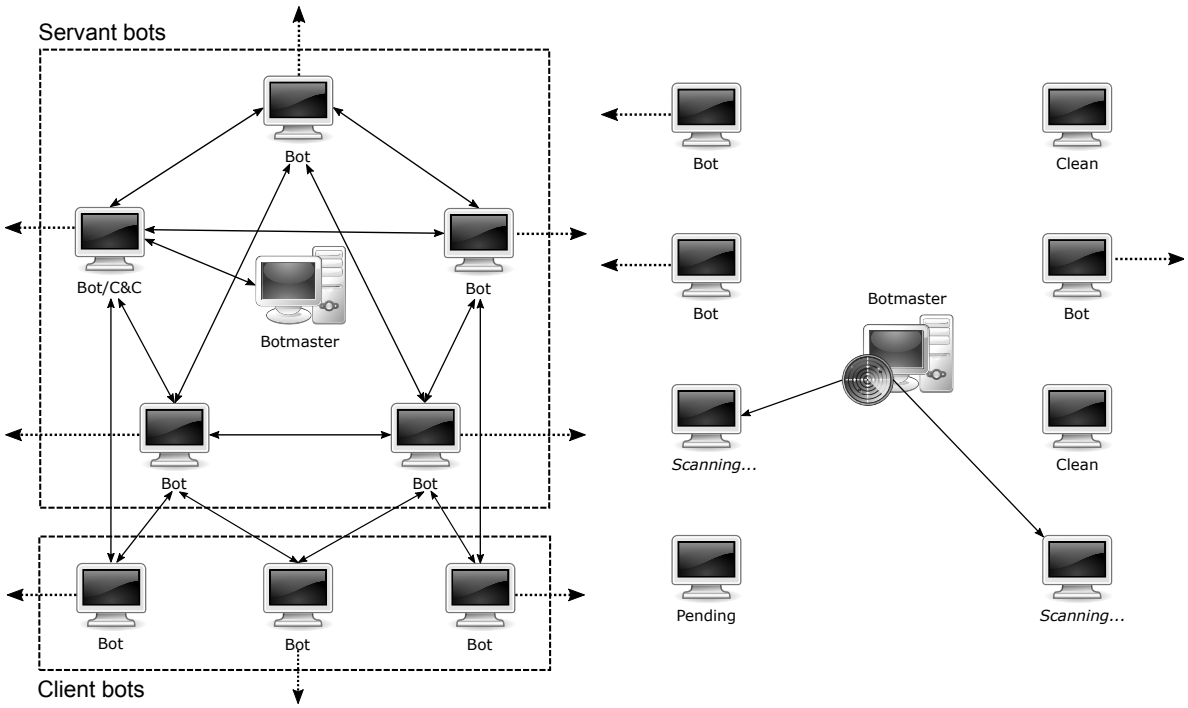
In this architecture a bot can act as a client as well as a server, eliminating a centralised point as the C&C server and, therefore, increasing the difficult to disarticulate the botnet.

- *Hybrid model C&C*: This model employs ideas from both centralised and decentralised schemes. It distinguished two types of bots. On the one hand, those who can act as a client and server simultaneously. They are known as *servant bots*, and have a public network address accessible from the Internet. These kind of bots are the only ones that can belong to peer lists. On the other hand, the client bots have a dynamic or private network address. All client bots must periodically connect to a servant bot in accordance to their peer lists, and with the aim of receiving new commands from the botmaster. Also, when a bot receives a new command that it has not been previously received, it must quickly forward the command to all servant bots on its peer list.

- *Random model C&C*. This scheme was introduced by Cooke *et al.* in [48] as a theoretical model, and goes a step further by increasing the difficult to disarticulate the whole botnet. The idea behind this strategy is that bots do not contact to the botmaster or other bots. Instead, they wait that the botmaster establish a connection against them. This implies that the botmaster must scan the entire network with the purpose of finding zombies and, if one is contacted, commands are sent to the bot. As a consequence of this process of discovering of bots, the architecture suffers from scalability and coordination problems.

2.4 Anonymous communications and Tor

The concept of anonymous communication was first introduced in 1981 by Chaum in his seminal paper "Untraceable electronic mail, return addresses, and digital pseu-donyms" [40]. Chaum proposed the use of *mix networks* (or *mixnets*), where the messages of a set of users are sent through a sequence of trusted



(a) Hybrid model C&C **(b)** Random model C&C

Figure 2.5: Architectures of the botnets (2/2)

nodes called *mix nodes* (*cf.* Figure 2.6). A mix node is basically a message relay that accepts batches of encrypted messages, decrypts, pads, and randomly permutes them to finally send them towards the next mix in each sequence. In such a way, mixes destroy the correlation between incoming and outgoing traffic. One important point is that messages are wrapped in several layers of encryption (one per mix), where the public key of each node is used. Moreover, and since the user messages are batched to create an anonymity set, the process can leads to a non-negligible delay. Consequently, mixnets are often considered anonymity networks designed for carrying unidirectional high-latency traffic in connectionless messages [139].

Afterwards, the idea of Chaum was extended by Goldschlag *et al.* under the notion of *Onion Routing* [68]. In this protocol, a user chooses randomly an ordered subset of nodes (named *onion routers*), in such a way that the selected nodes will route the traffic over the formed path (a *circuit*). In order to provide anonymity to the user, each router is only able to know its predecessor and its successor in the circuit. Moreover, each onion router does not know how many nodes the circuit has, nor its position unless it is the last. This properties are mainly achieved by means of sending the original messages of the user in the form of an *onion* (*cf.* Figure 2.7), *i.e.* a wrapped message in multiple layers of encryption (one layer for each node in the circuit). When a node receives a message, it peels off its layer of encryption and sends the resulting value to the next node. The process is repeated until the last router of the circuit, where the original message is recovered and forwarded toward the real destination server. One can observe that, unlike mixnets, an anonymity set is not built in each router and, therefore, there is not any specific delay for such purpose. In addition, the key point behind this construction is how a

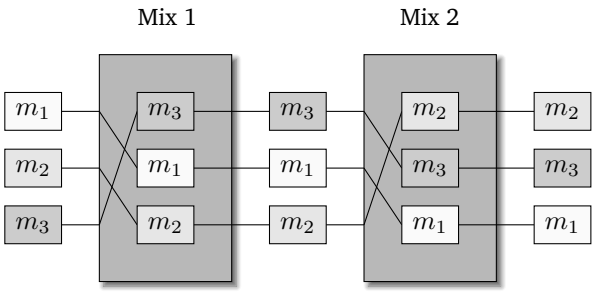


Figure 2.6: A mixnet with two mix nodes

secure channel is established through each onion router. Goldschlag proposed that the user sends to each onion router an encrypted message using its corresponding public keys. Such encrypted message contains a random symmetric key used to encrypt the corresponding layer along with the name of the next node in the circuit.

Tor is the second generation of onion routing protocol proposed by Dingledine *et al.* [52], which resulted in the implementation of one of the most widely-used low latency solutions. Their proposal addresses some limitations in the original Onion Routing. One of this enhancements is to replace the asymmetric encryption with much more efficient symmetric cipher to improve the onion routing. The strategy proposed by Dingledine relies on using the public keys of the routers in order to establish temporary session keys via an interactive Diffie-Hellman protocol. In particular, the session keys are agreed during an initial phase called *circuit construction*, where a public-key infrastructure in conjunction with of a set of *directory servers* is used. The circuit construction phase uses a technique called *telescoping* where the circuit is built incrementally, negotiating a symmetric key with each onion routing on the circuit, one hop at a time. Such protocol is known as The Authentication Protocol (TAP) and was formally proven secure by Goldberg [66].

Although other proposals have improved the efficiency of the telescoping method used in TAP, the main drawback of these solutions is the degree of interchanged messages. Indeed, to build a circuit composed by n routers adopting a telescopic strategy, it is required the exchange $n(n+1)$ symmetrically encrypted messages, which implies a complexity of $\mathcal{O}(n^2)$. Some published papers have improved this by means of applying different methods such as identity-based cryptography, Certificateless Public Key Encryption, or Diffie-Hellman Chains among others. All of these schemes relies on improving the establishment of the shared keys between the client and each node of a circuit. However, these proposals introduce scalability and security drawbacks.

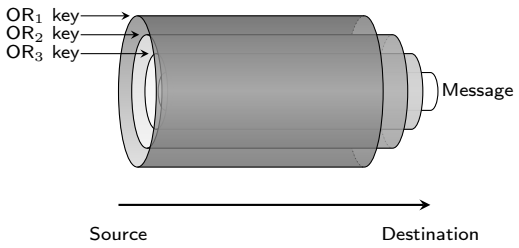


Figure 2.7: Conceptual representation of encryption layers in Onion Routing

2.5 Conclusion

In this chapter we have presented some basic concepts that will be used in the rest of this dissertation, and which comprises: (1) An introduction of some mathematical notation, models, and core definitions that are necessary to understand the rationale of our work, (2) A brief description of the DNS threats and the DNSSEC extension, (3) An overview of the botnets, including the life cycle of zombies and the botnet architectures, and (4) An introduction to the Tor anonymous network.

3

DNS and fast-flux networks

> *If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.*
>
> SUN TZU

In the context of the botnets, fast-flux refers to the strategy of hiding the C&C servers. Such servers are crucial for the life cycle of the botnet (*cf.* Section 2.3.1). The idea behind this technique is to have multiple IP addresses associated with a fully qualified domain name, where the IP addresses are changed with high frequency through the modification of the DNS records. Behind each IP associated there is a *flux agent* that acts as a proxy forwarding the traffic towards the C&C servers called *motherships*. The *motherships* are responsible of managing the DNS infrastructure related with the fast-flux domain, controlling the bots, and serving the contents. This way, botnet operators increase the robustness of their C&C services by making the botnets much more resilient against countermeasures and failures of individual proxy nodes. However, and for the very same reason, the discovery of fast-flux services and their associated resources is a valuable way to discover botnet activities during its life cycle. In this chapter we present a novel approach for the detection of fast-flux domains in real time that reduces the likelihood of erroneous detection, while providing better results than previous research efforts.

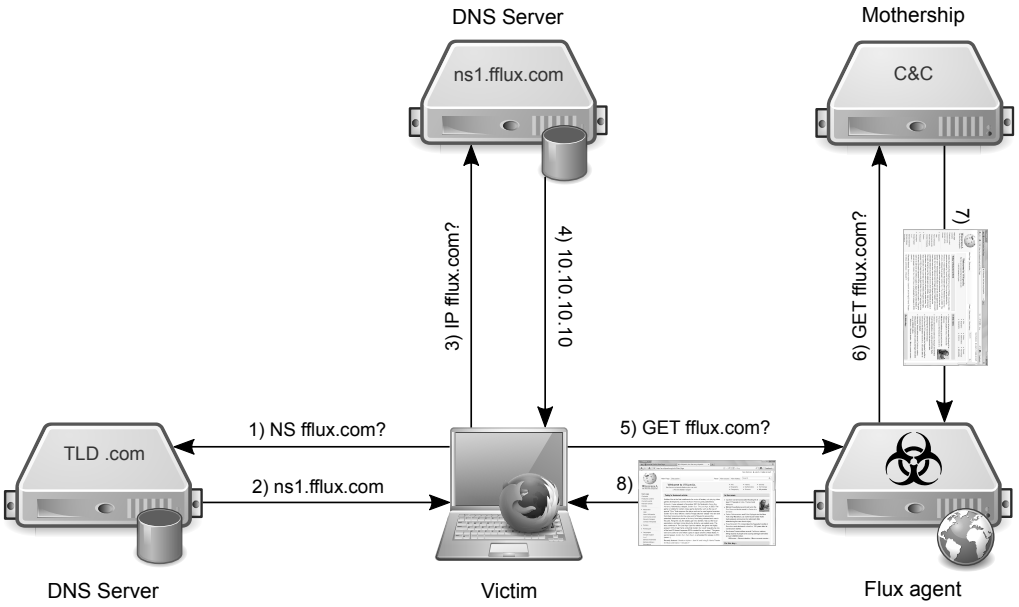


Figure 3.1: Conceptual representation of a single flast-flux network

3.1 Fast-flux architectures and mitigation techniques

In this section, we describe the the fast-flux service networks and its operation; in particular, the single and double fast-flux architectures are depicted. Also, we overview the state of the art regarding the mitigation strategies that we could adopt against such networks.

3.1.1 Fast-flux architectures

In the context of fast-flux service networks we have two main architectures: the single and the double. The goal of both architectures is the same: to increase the robustness of the C&C services making the botnets more resilient against countermeasures and failures of individual proxy nodes. Moreover, this technique is used to difficult the identification (*i.e.*, the IP address), and to frustrate location and shutting down of servers used for illegal activities.

The most simple architecture is known as *single fast-flux*. Figure 3.1 depicts a conceptual representation of this first technique. Associated to this architecture there is a fully qualified domain name (also known as *flux domain*) linked to multiple IP addresses that are constantly changing by the modification of the DNS A records. This is also accomplished by means of registering the domain with a short TTL (Time-To-Live). Taking down malicious DNS records is more difficult than a compromised system with a single IP, since many DNS records can be established for many IP addresses. Behind each IP associated to the flux domain there is a compromised computer (or *flux agent*) that acts as a proxy. Thus, a request from a victim to the flux domain will go through one flux agent before being forwarded to the backend server *(*or *mothership*). In this manner, single fast-flux networks include an abstraction layer that increases anonymity, availability, load balancing and resiliency to takedown.

The second architecture, or *double fast-flux*, provides an additional layer of redundancy compared to

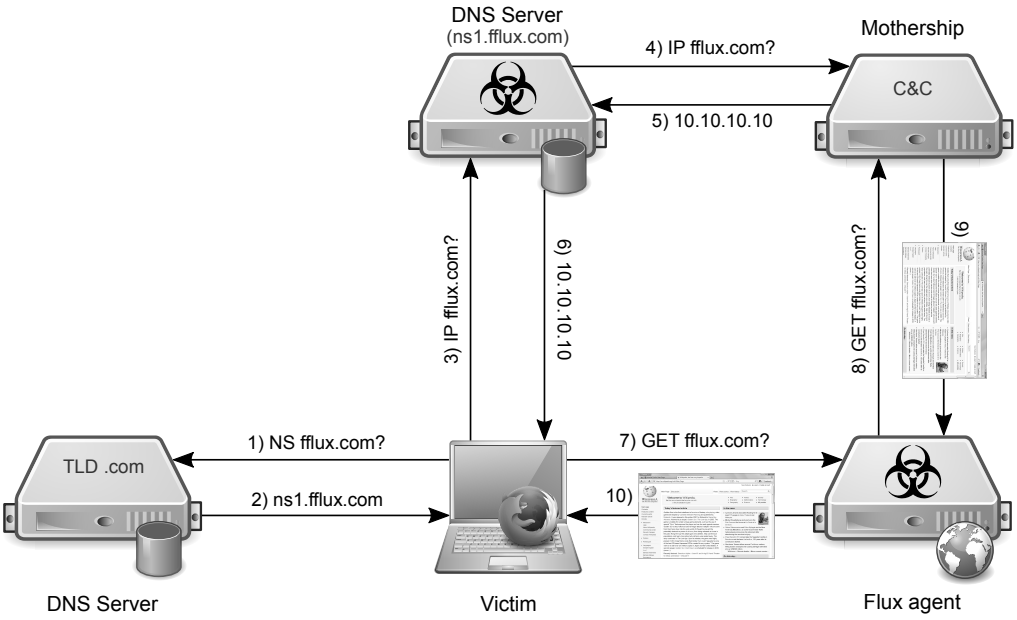


Figure 3.2: Conceptual representation of a double flast-flux network

the single flast-flux. Figure 3.2 depicts this architecture. This additional layer of redundancy is achieved by not only changing continually the DNS A records, but also the authoritative NS records related to the malicious domain. In this manner, fast-flux operators can associate the IP addresses of the authoritative DNS servers to compromised systems. Then, when a DNS request for the flux domain is received from the client, the current authoritative name server forwards the queries to the mothership node for the required information.

3.1.2 Mitigation techniques

From the standpoint of the mitigation strategies, it seems that there is not effective techniques beyond real-time detection methods. The use of DNSSEC extension presented in Chapter 2, which could seem a promising strategy, is not as efficacious as we could think. Firstly, because the DNSSEC extension is still not widely deployed, and secondly because under a scenario like the double fast-flux the Authoritative Domain Servers are controlled by the fast-flux operators.

In [73], Holz *et al.* suggest the mitigation of fast-flux networks by the creation of *domain blacklists* in collaboration with (1) domain name registrars, who has the authority to shut down a domain, and (2) Internet Service Providers (ISP), who can blackholing DNS requests for fast-flux domains included in such domain blacklists. Another proposal by Holz *et al.* is that ISPs change their policies and block certain incoming connections requests directed to dial-up IP ranges (*e.g.* TCP port 80, or UDP port 53) since users behind this IPs does not need to host this services. We believe that the strategies presented by Holz *et al.* are not as effective as we would expect. On the one hand, and according to Konte *et al.* [87], the dynamics of fast-flux service networks make ineffective mitigation schemes that relies on blacklisting. Operators of fast-flux domains can swap out the blacklists hosts, or use another techniques such as the use of Domain Generator Algorithms (DGA) [135]. On the other hand, it seems infeasible

that all the ISPs around the world use the same block policy and, therefore, if an operator of a fast-flux domain detects that the IP of a flux-agent belongs to a ISP that applies such blocking policy, he will simply change the port (when possible), or excludes this system from its fast-flux network. For these reasons, we believe that real-time detection strategies —and the subsequent blocking— should be the cornerstone to mitigate fast-flux domains.

3.2 Related work

The detection of fast-flux service networks is a hot research topic. There is a great number of approaches for malicious fast-flux detection, ranging from training classifiers, such as [73, 75, 95, 155, 158], to collaborative systems, such as [162, 163, 164]. To our knowledge, most relevant proposals in the literature are machine learning based. A relevant approach in this category is the work of McGrath *et al.* presented in [95]. The authors build a linear classifier grounded on Support Vector Machines (SVM) [137], and define a minimum set of features required to detect a fast-flux domain. Such features are the number of IP addresses associated to a given domain, the number of ASN (Autonomous System Numbers), the number of different prefixes, and the number of different countries that the associated IP addresses belong to. Hsu *et al.* presented in [75] an enhanced SVM classifier, whose detection features are completely different to those of McGrath *et al.* work. The new classifier bases their features in the intrinsic characteristics that bots have, such as the network delay, the request processing delay, and the document fetching delay.

3.3 Our detection proposal

A fast-flux detector system must provide real-time decisions. This way, it is possible to warn potential victims before they connect to a malicious site. Most existing proposals in the literature rely on the number of IP addresses by querying a certain domain name, or by passively monitoring DNS queries, for a certain period of time. Thus, the time required to detect a fast-flux domain with such strategies is counterproductive. At the same time, an efficient fast-flux detector system should minimise the number of erroneous detections (*i.e.*, both false positive and negative rates). Erroneous detections are often caused by the similarities between illicit fast-flux network systems and similar (legitimate) services such as Round Robin DNS and Content Delivery Networks. The goal of our proposal is twofold: (i) to provide a real-time detection strategy which does not require a long period of time for the detection, and (ii) to prevent erroneous recognition of legitimate DNS-based services that can be flagged as malicious fast-flux domains by mistake.

We propose the construction of a novel linear SVM classifier that extends those of Hsu *et al.* and McGrath *et al.* presented, respectively, in [75, 95]. These two approaches based their detection properties on the definition of certain fast-flux features. The main drawback of the McGrath *et al.* classifier is that it can be misled by botmasters, due to the nature of its set of detection features. For instance, the botmaster can assign less IP addresses to a domain, or use heuristics to select only those bots geolocated in the same country [85]. This would lead the McGrath *et al.* classifier to a great rate of false negatives. Contrarily, the set of features of the Hsu *et al.* classifier are intrinsic to malicious fast-flux networks and cannot be manipulated by the botmaster. Nevertheless, it also presents an important drawback. It can

be misled by legitimate servers, *e.g.*, Round Robin DNS servers or Content Delivery Network servers, and end with a great number of false positives.

Our proposal addresses these two aforementioned limitations. First, it differentiates malicious fast-flux networks by their own features. It does so in an automatic way by using machine learning techniques to build a new SVM classifier trained via real features extracted from domains and bots. Below, we present our proposed set of detection features.

3.3.1 Detection features

Applying the features from [75] and [95] separately leads to false positives and false negatives. We propose to merge both kind of features with the aim of reducing false detection rates. The rationale is that those false positives and false negatives caused by the features of the first classifier shall be countered by the features from the second classifier, and vice versa. Based on this idea, we build a new set of features. The set can be divided in two different groups: (1) DNS-related features and (2) bot-intrinsic features. The former being features that are related to the DNS resolution process. The latter being features that are inherent to infected computers. In the sequel, we detail each of the feature sets.

DNS-related features

Our proposed set of DNS-related features contains those characteristics that can be obtained by using information about DNS. The information is extracted by using a DNS request issued to the authoritative name server for a given domain, and then is processed to obtain the features. The features that are contained in this group are the minimum set of DNS-related features needed to detect fast-flux [95]. We describe some sample DNS-related features next:

- **Number of IP addresses associated to the same domain:** Conventional legitimate domains usually have either one or two IP addresses associated to them. In fast-flux networks (legitimate or not) and similar technologies, such as CDNs (Content Delivery Networks) or RRDNS (Round Robin DNS), the number of associated IPs tends to be much higher. In fact, fast-flux, CDNs and RRDNS use multiple IP addresses for a given domain, with the goal of providing high availability and greater performance to the end user. This feature, then, tends to discriminate those conventional legitimate domains from fast-flux networks, CDNs, and RRDNS.

- **Number of associated Autonomous System Numbers:** The servers associated to a given conventional legitimate domain are usually located within the same autonomous system, as they are usually managed by the same company. In botnets, however, this is not the case. They are composed by infected domestic host computers that are spread across the world. Regarding CDNs or RDDNS, they exhibit the same behaviour as legitimate domains, *i.e.*, appear as a single vantage point within a unique autonomous system.

- **Number of associated prefixes:** IP address prefixes also give information about whether a domain is either legitimate or part of a fast-flux service. The IP addresses of hosts of legitimate networks usually belong to a few BGP prefixes per hostname, while in networks exhibiting fast-flux are usually associated to multiple BGP prefixes per hostname.

- **Number of associated countries:** As in the case of the Autonomous System Numbers, the servers associated to a legitimate domain are typically located within the same country. In fact, hosts belonging to a particular country code TLD (Top Level Domain) are typically located on IP addresses physically residing within that country. However, hosts of fast-flux domains are typically spread around the world. Therefore, a hostname associated to multiple countries is likely to be part of a fast-flux service.

Bot-intrinsic Features

The bot-intrinsic features are those strongly related to the characteristics of the compromised machines, that is, the bots. In this group of features we assume that botnet owners exploit the bots to execute web-based malicious services such as phishing pages and malware delivery sites. Therefore, the malicious software operating on each bot is assumed to provide an HTTP service and related flows.

It is importat to remark that —as we introduced in Chapter 2— botnets are typically formed by malware-infected home computers. Usually, there are big differences in hardware and software between home computers and dedicated hosting servers. Dedicated hosting servers are much more powerful, and connected to Internet via high bandwidth connections in order to obtain the best possible performance. Their running processes are those dedicated to provide web services. On the contrary, home computers have a more limited hardware, the bandwidth of their connection is also much more limited, and they run all kinds of software. These differences can be used to extract features to help discriminate legitimate domains from fast-flux domains. We describe some sample bot-intrinsic features next:

- **Network delay:** Refers to the time required to transmit packets back and forth over the Internet between a client and a server. It can be obtained by computing the difference between the time a client sends out the first `TCP SYN` packet to the server and the time the client receives the corresponding `TCP SYN+ACK` packet from the server.

- **Processing delay:** Refers to the time required for the server to process an erroneous HTTP request that does not incur any additional computation and I/O operations. Its measurement is done by sending out an HTTP request with an undefined method, such as a nonsense `BADMETHOD` method, and computing the difference between the sending time of the non valid request and the time when a 400 (Bad request) or 405 (Method Not Allowed) response is received. Then, the network delay has to be subtracted to this value, in order to obtain the processing delay.

- **Document fetch delay:** Refers to the time required from the server to fetch a web page, either from a hard disk or from a backend mothership. The fetch operation occurs at the server side and we cannot know exactly what happens, we compute it by doing the following. We compute the time difference between the send out of an `HTTP GET` request and the time the client receives the corresponding HTTP response (200 OK), and then subtracting the network delay. This way, we obtain an estimator of the document fetch delay.

3.3.2 Building the linear SVM classifier

In order to detect fast-flux behaviour we build a linear classifier by using the features presented in Section 3.3.1. The linear classifier used is based on Support Vector Machines (SVM) [137], which

is a non-probabilistic binary classifier that constructs a hyperplane in a very high-dimensional space, achieving this way a good separation when the hyperplane has the largest possible distance to the nearest training data points.

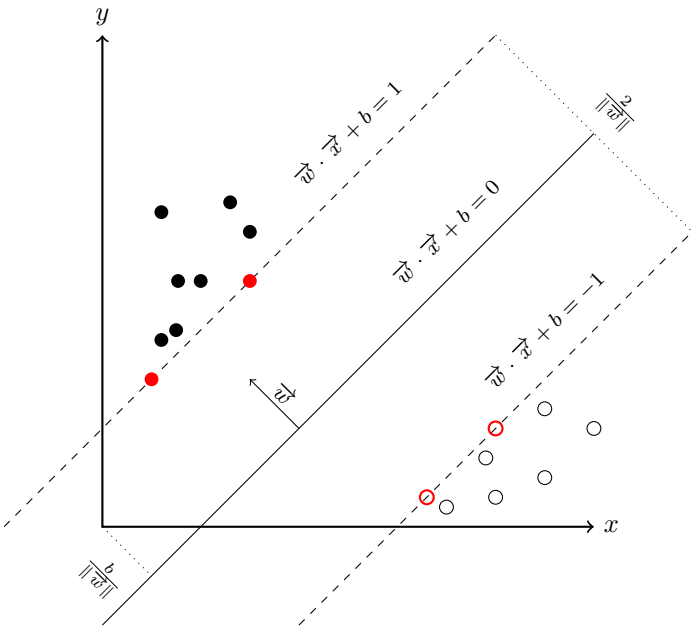


Figure 3.3: Conceptual representation of a linear SVM building in $\mathbb{R}^2$

More formally, we are given a training dataset $\mathcal{D}$ represented as a set of n points of the form described in Equation (3.1). These points —denoted by $\vec{x}_i$— are p-dimensional real vectors where each coordinate represents a characteristic, and $y_i = 1$ or $y_i = -1$ is the class to which they belong to. The goal is to find the hyperplane that divides the group of points $\vec{x}_i$ for which $y_i = 1$ from the group of points for which $y_i = -1$, in such a way that the distance between the hyperplane and the nearest point $\vec{x}_i$ from either group is maximised.

$$\mathcal{D} = \{(\vec{x}_i, y_i) \mid \vec{x}_i \in \mathbb{R}^p,\ y \in \{-1, 1\}\}_{i=1}^{n} \tag{3.1}$$

When the data to be classified is linearly separable, as it is our case, two parallel hyperplanes can be computed so that they separate the two classes and, at the same time, the margin between them is the maximum possible. Those hyperplanes can be described by the equations:

$$\vec{w} \cdot \vec{x} + b = 1$$
$$\vec{w} \cdot \vec{x} + b = -1$$

and the margin between them is given by the expression $\frac{2}{||\vec{w}||}$ (*cf.* Figure 3.3). Also, to prevent data points from falling into the margin, two constraints are defined:

$$\vec{w} \cdot \vec{x} + b \geq 1, \quad \text{if } y_i = 1$$
$$\vec{w} \cdot \vec{x} + b \leq -1, \quad \text{if } y_i = -1$$

that can be simplified in the following single expression:

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, \quad i = 1, ..., n$$

Therefore, the problem to obtain the classifier can be written as:

$$\begin{aligned} \max_{\vec{w},b} \quad & \frac{2}{||\vec{w}||} \\ \text{s.t.} \quad & y_i(\vec{w}\cdot\vec{x}_i - b) \geq 1, \quad i = 1, ..., n \end{aligned}$$

which can be rewritten in an equivalent quadratic optimisation problem known as the *primal form*:

$$\begin{aligned} \min_{\vec{w},b} \quad & \frac{1}{2}||\vec{w}||^2 \\ \text{s.t.} \quad & y_i(\vec{w}\cdot\vec{x}_i - b) \geq 1, \quad i = 1, ..., n \end{aligned}$$

In order to solve this equivalent problem, a new more convenient form called the *dual form* can be derived by means of the Lagrange multipliers method. By applying this strategy the constrains are eliminated as follows:

$$\max_{\vec{\alpha};\alpha_i \geq 0} \min_{\vec{w},b} L(\vec{w}, b, \vec{\alpha})$$

where $L(\vec{w}, b, \vec{\alpha})$ is the Lagrange function defined by:

$$L(\vec{w}, b, \vec{\alpha}) = \underbrace{f(\vec{w}, b)}_{\text{objective function}} + \sum_{i=1}^{n} \alpha_i \underbrace{g_i(\vec{w}, b)}_{\text{inequality constrains}}$$

Thus, by applying the objective function $f(\vec{w}, b) = \frac{1}{2}||\vec{w}||^2$ and the constrains $g_i(\vec{w}, b) = y_i(\vec{w}\cdot\vec{x}_i + b) - 1 \geq 0$ to the Lagrange function we obtain:

$$L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2}||\vec{w}||^2 + \sum_{i=1}^{n} \alpha_i \big[y_i(\vec{w}\cdot\vec{x}_i + b) - 1\big] \tag{3.2}$$

Then, if we take into consideration that the Lagrange function is convex, it is possible to find the minimum value by imposing the restriction that partial derivatives are equal to zero. That is:

$$\frac{\partial L}{\partial \vec{w}} = 0 \rightarrow \vec{w} - \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i = 0 \rightarrow \vec{w} = \sum_{i=0}^{n} \alpha_i y_i \vec{x}_i$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

and replacing these values in Equation 3.2 we obtain:

$$\min_{\vec{w},b} L(\vec{w}, b, \vec{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \vec{x}_i \cdot \vec{x}_j$$

Now, by maximising this expression with respect to $\vec{\alpha}$ and applying the constraint we derived from the partial derivatives, we get the final dual form which is a problem less complex to solve:

$$\begin{aligned} \max_{\vec{\alpha}} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \vec{x}_i \cdot \vec{x}_j \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0 \end{aligned}$$

Domain	#IP	#ASN	#PREF	#C	ND	PD	DFD	Label
adultdatinghouse.info	3	3	3	3	0.1362	1.5847	1.5117	malicious
google.com	11	1	1	1	0.0481	0.0280	0.2593	legitimate
cosmodatelab.info	3	3	3	3	0.1160	0.2609	1.1189	malicious
cupidlocals.com	2	2	2	2	0.4668	0.1136	0.1718	malicious
youtube.com	11	1	1	1	0.0460	0.0314	0.3419	legitimate
yahoo.com	3	3	3	1	0.2443	0.3965	0.5745	legitimate

Table 3.1: Data set example

3.4 Experiments

To validate our SVM classifier, we first retrieve some sample lists of malicious fast-flux domains as well as legitimate domains. We use these lists to extract the features presented in Section 3.3 and label them accordingly (*i.e.*, labelling each domain as malicious or legitimate). We obtained a list of 81 active malicious fast-flux domains from the *Atlas* Web site [9], and a list of 81 legitimate active benign domains from the *Alexa Top Sites* site [4]. These lists were processed in order to extract the features. For that purpose, we built a set of scripts written in Python. To obtain the different DNS related features, the script issues DNS requests in order to obtain the NS records as well as the A records. Then, using the IP addresses obtained with the requests and connecting to the *cymru* whois service [142], their respective autonomous systems, countries and prefixes are obtained. To obtain the bot-intrinsic features, the script sends a `TCP SYN` packet in order to measure the network delay, an `HTTP GET` request to measure the document fetch delay, and an invalid `HTTP BADMETHOD` request to measure the processing delay. The measurements of the delays are repeated 10 times and then the average is computed. An example of the data set can be found in Table 3.1.

Once the data sets have been collected, the SVM classifier —written in Java and using the Java-ML library[1]— is validated by using the k-fold cross-validation method [86]. With this method, the data sets are split in k different groups, using one of them to train the classifier, and the other $k-1$ to classify. This is repeated other $k-1$ times, so that every time the training is done with a different group. The results obtained from these experiments are presented next.

3.4.1 Obtained results

Table 3.2 outlines the number of true positives, false positives, true negatives, and false negatives for the three evaluated classifiers. Notice that our proposal obtains better results than just using the features from [75] and [95] separately. Therefore, and as expected, by combining the two kind of features we noticeably reduce the false positives and the false negatives, at the same time that we slightly increase the true positives and true negatives. One can observe that our proposal provides only one misclassified results, which happens to be a false positive. Thus, all the malicious fast-flux domains were correctly detected, and only one legitimate domain is misclassified as a malicious fast-flux domain from our classifier. In order to clarify the weaknesses of the strategies proposed by McGrath and Hsu, we analysed their corresponding false positives and negatives obtained in our experimental results. We discovered that such misclassifications were deeply tied with the features used by them, and not with

	Hsu [75]	**McGrath** [95]	**Our proposal**
True Positive	63	78	81
False Positive	9	4	1
True Negative	72	77	80
False Negative	18	3	0

Table 3.2: Comparative study between [75], [95] and our proposal

the classifier itself.

On one hand, the McGrath's classifier exhibited only one false positive that was caused by a legitimate domain with a set of seven IP addresses associated to six different autonomous systems, seven different prefixes, and distributed along five different countries. This is not a common characterisation of a legitimate domain, and this was the reason why the classifier considered it as a fast-flux domain. With respect to the false negatives, all the malicious domains classified erroneously had a reduced number of IP addresses, autonomous systems, prefixes and countries. This confirms our hypothesis that a malicious botmaster can create a fast-flux domain with a particular chosen set of features, and whose aim is to evade any detection based on using only DNS-related information.

On the other hand, the false positives obtained after using the Hsu proposal were caused by high values of network delays, processing delays and document fetch delays. Probably, such values were a consequence of a network problem (*e.g.* network congestion) or a high resource consumption from the server side. This corroborates that by using the bot-intrinsic features in an isolated manner can not be sufficient for an efficient detection, since some random perturbation from the network or server standpoint can lead to undesirable misclassifications. From the false negatives point of view, we observed that the delay values associated to such fast-flux domains were low enough to consider them as benign domains. Again, we can argue that the botmaster in charge of those fast-flux domains constructed them by using bots with a good network and server resources, leading to an evasion mechanism.

After analysing how a malicious fast-flux domain could evade the proposals of McGrath and Hsu independently, one could think that the unification of both evasion strategies could be applied in order to elude our classifier. Although this is theoretically possible, our experimental results have shown that none of the studied domains has revealed this behaviour. We believe that the combination of features increases notably the difficulty of constructing a malicious fast-flux domain that is able to evade the detection. Moreover, our proposal has proved to be more robust against false positives introduced by legitimate values related to the DNS features, or by random noise in the delays measurements. This is possible since the DNS features counterbalance the bot-intrinsic features –or vice versa– in case of a false positive related with one of both scopes.

3.5 Conclusion

Botnets use fast-flux as an evasion strategy to make difficult the trace-back and posterior take down. For such purposes, fast-flux operators exploit the lack of security mechanisms that the DNS protocol has, and that we have introduced in Chapter 2. Despite that, there are also legitimate fast-flux networks, as well as similar technologies such as round robin DNS and content delivery networks. Detecting malicious

fast-flux networks implies being able to discriminate them among those similar technologies. Most approaches in the literature use detection features that may mislead the discovery process and end with high rates of false positives and false negatives. In this chapter we have extended two existing classifiers based on SVM (*Support Vector Machine*) that suffer from such limitations, and conducted simulations that verify the feasibility and superiority of our approach.

4

Privacy in DNS with Tor and PIR

> *For the first time he perceived that if you want to keep a secret you must also hide it from yourself. You must know all the while that it is there, but until it is needed you must never let it emerge into your consciousness in any shape that could be given a name.*
>
> GEORGE ORWELL

In previous chapters we have seen how the DNS protocol suffers from several weaknesses that put in risk the security of its users. The lack of security of the DNS protocol when it was conceived, together with the emergence of new technologies, leads us to deal with new challenges. This chapter comes from privacy and security concerns regarding the use of the protocol DNS as the underlying mechanism of new Internet protocols, such as the ENUM (*tElephone NUmber Mapping*) service. ENUM is indeed a set of service protocols used on VoIP (Voice over IP) applications. One of the main characteristics of ENUM is the mapping of traditional phone numbers associated to the ITU-T (International Telecommunications Union) E.164 recommendation, to URIs (Universal Resource Identifiers) from VoIP providers, as well as to other Internet-based services, such as e-mail, Web pages, etc. We overview in this chapter some of the features of this service, as well as some security and privacy concerns regarding the use of the DNS protocol in ENUM, and the threats introduced in Chapter 2.

4.1 The ENUM service

The ENUM service is a suite of protocols used in VoIP applications whose main goal is the unification of the traditional telephone E.164 system with the IP network of the Internet. Designed and developed by the *Internet Engineering Task Force* (IETF) in late nineties, ENUM allows the mapping of IP services by using an indirect lookup method based on DNS technologies. In this manner, and by simply using existing DNS implementations, ENUM allows retrieving lists of IP based services, such as SIP (*Session Initiation Protocol*) identifiers for VoIP applications, e-mail addresses, Web pages, etc., associated to the principal of an E.164 telephone number. ENUM uses a particular type of DNS records, called Naming Authority Pointer (NAPTR) [96]. Instead of resolving host or service names into IP addresses, the ENUM service translates E.164 telephone numbers into Uniform Resource Locators (URLs) embedded within NAPTR records. At long term, ENUM is expected to become a decentralised alternative to the E.164 system. For a more detailed introduction to the suite of protocols associated with ENUM, we refer the reader to [72].

As a matter of fact, ENUM is just a simple convention for the translation of E.164 telephone numbers, such as +1-012345678, into URI (*Uniform Resource Identifier*) strings. These strings are associated to the DNS system by using the following convention: (1) special symbols like '+' and '-' are deleted (*e.g.*, +1-012345678 becomes 1012345678); (2) the resulting string of digits is inverted from left to right (*e.g.*, 8765432101); (3) a symbol '.' is inserted between each two digits (*e.g.*, 8.7.6.5.4.3.2.1.0.4.1); (4) the domain name .e164.arpa (registered by the IETF for ENUM resolution) is finally concatenated to the previous string (*e.g.*, 8.7.6.5.4.3.2.1.0.1.e164.arpa). The resulting string of characters and digits is then ready to be used as a normal query towards the DNS system. At the server side, the URI associated to every possible telephone number registered by ENUM is stored together with information about its principal (*e.g.*, owners or users of those telephone numbers). Such an information is stored on DNS records of type NAPTR. The internal structure of these records offers to ENUM enough storage space and flexibility for managing complex information (*e.g.*, use of regular expressions).

Let us show in the following a complete example in which ENUM is used for the translation of the telephone number +1-012345678 associated to a user U_1. Let us assume that a user U_2 wants to get in contact with user U_1. First of all, user U_2 translates the previous telephone number into the string 8.7.6.5.4.3.2.1.0.1.e164.arpa. U_2 then uses the obtained URI to construct a DNS query of type NAPTR by using the command line tool *dig*:

```
dig @$NS -t NAPTR 8.7.6.5.4.3.2.1.0.1.e164.arpa
```

As a result, U_2 obtains the information included in Table 4.1:

Order	Pref.	Flags	Service	Regexp.	Replacement
100	10	u	sip+E2U	!^.*$!sip:u1@sip.com!'	.
101	10	u	mailto+E2U	!^.*$!mailto:u1@mail.com!	.
102	10	u	http+E2U	!^.*$!http://www.u1.com!	.
103	10	u	tel+E2U	!^.*$!tel:+1-012355567!	.

Table 4.1: Example of an ENUM DNS response to a given query

Let us analyse the response returned by *dig*. As we introduced above, NAPTR records support the use of regular expression pattern matching [96]. In case a series of regular expressions from distinct NAPTR records need to be applied consecutively to an input, the field *Order* is used. The value given in the first line, set to 100, indicates that from the four results of the query, the service *SIP* has the highest priority. In case of having more that one record with the same *order* values, the following field, *i.e.*, *Pref.*, decides which information must be used first. The field *Flag* given for each line, and set to the value *u*, indicates that the field *Regexp.* associated with every record contains the URI associated to the requested E.164 telephone number. A field *Replacement* containing the operator *'.'* indicates to the ENUM client of user U_2 that the final URL is indeed the string placed between the markers *'!^.*$!'* and *'!'* of the expression contained within the field *Regexp*. The field *Service* indicates the kind of IP service that can be found in the resulting URL. For example, the field *Service* associated with the first line indicates that the resulting service is based on the SIP protocol [77]. The other three options returned as a result of the query are (1) an e-mail address associated with user U_1, (2) his personal Web page, and (3) the use of an additional E.164 telephone number.

Let us notice from our example that the ENUM service does not resolve the IP addresses associated to the URLs embedded within the NAPTR records. A DNS query of type 'A' must follow after an ENUM resolution with the objective of resolving the appropriate IP address that will eventually be used to contact the final service. In our example, and given the values of the field *Order* discussed above, user U_2 contacts again the DNS server in order to obtain the IP address associated to the SIP at `sip.u1.com` to request the connection to user U_1 (*i.e.*, `u1@sip.u1.com`).

4.2 Threats to the ENUM service

The use of the DNS protocol as the underlying mechanism of the ENUM service leads to security and privacy implications. The exploitation of well known vulnerabilities of DNS-based procedures is a clear way of attacking the ENUM service. An analysis of critical threats to ENUM may be found in [116, 117]. Rossebø *et al.* present in these works their risk assessment analysis of the ENUM service based on a methodology proposed by the European Telecommunications Standards Institute (ETSI). Both threats and vulnerabilities reported in these works are indeed a heritage of the vulnerabilities existing in DNS mechanisms. We can find in [12] a complete analysis of threats to DNS technologies. The most important threats to DNS technologies can be grouped as follows: (1) authenticity and integrity threats to the trustworthy communication between resolvers and servers; (2) availability threats by means of already existing denial of service attacks; (3) escalation of privilege due to software vulnerabilities in server implementations. Moreover, the DNS protocol uses clear text operations, which means that either a passive attack, such as eavesdropping, or an active attack, such as Man-in-the-Middle, can be carried out by unauthorised users to capture queries and responses. Although this can be considered as acceptable for the resolution of host names on Web services, an associated loss of privacy when using DNS for the resolution of ENUM queries is reported in [116, 117] as a critical threat. The security extensions of DNS, known as DNSSEC (*cf.* Chapter 2, Section 2.2.1, for further details), can mitigate some of these threats —in particular those related to authenticity and integrity— but not all of them.

We consider that the loss of privacy in ENUM queries is an important concern. Beyond the engineering advance that the ENUM service supposes, it is worth considering the consequences that the

exposure of people's information may suppose. The achievement of such information by dishonest parties exploiting flaws and weaknesses in the service itself or its underlying protocols must be avoided. We can consider, for instance, worst case scenarios where dishonest servers associated to unscrupulous service providers start keeping statistics of ENUM queries and building people's profiles based on their communication patterns [160]. These scenarios may lead to further violations, such as spam, scams, untruthful marketing, etc. Consumers must be ensured that these activities are not possible [47]. In fact, some measures has been proposed by the IETF in order to reduce the risk from the point of view of the privacy. This includes the limitation of the personal information stored by ENUM, as well as to require to persons and institutions to sign some kind of consent statement before being included in ENUM databases. In spite of this, and beyond to limit the amount of personal information included in ENUM databases, no mechanisms seems to have been proposed in order to warranty the privacy of the queries performed by third parties.

4.3 Related work

A first solution to address the privacy concerns is the use of anonymous-based communication infrastructures. The use of strong anonymity infrastructures can suppose, however, a high increase of the latency of a service like the DNS and the ENUM services. We recall that a communication infrastructure for these services must ensure that the service itself is able to deliver both queries and responses accurately and in a timely fashion. Thus, strong anonymity does not seem to be compatible with this requirement. On the other hand, the use of low latency infrastructures, such as the anonymous infrastructure of the Tor (*The second generation Onion Router*) project [52], based in turn on the *Onion Routing* model [115], is more likely to meet the performance requirements of the DNS/ENUM service. Nevertheless, a solution based on both Tor and *Onion Routing* may only be useful for hiding the origin of the queries. Although by using such proposals senders are indeed able to hide their identities through a network of *proxies*, they do not offer anonymity to the queries themselves. For instance, threats due to the existence of dishonest servers are not covered by these solutions [62].

As an alternative, the approach presented by Zhao *et al.* in [160, 161] aims at preserving the anonymity of DNS/ENUM queries from the point of view of the channel and/or the service providers. The main objective of these proposals is the achievement of anonymity by using a PIR (Privacy Information Retrieval) model [107]. The authors propose devising the communication protocol involved between DNS clients and servers by considering queries as secrets. Instead of querying the server by a specific host name h, for example, Zhao *et al.* propose in [160] the construction and accomplishment of random sets of host names $[h_1, h_2, \ldots, h_n]$. The resulting protocol aims at avoiding that by listening into the channel or controlling the destination service, an attacker learns nothing about the specific host name h from the random list of names. The main benefit of this proposal is the simplicity of the approach. The main drawback is the increase in communication bandwidth that it may suppose. Zhao *et al.* extend in [161] this first proposal towards a two-servers PIR model. The objective of the new protocol is to guarantee that DNS clients can resolve a given query, at the same time that they hide it to each one of the servers. Nevertheless, compared with the previous proposal, this approach reduces the bandwidth consumption. The approach requires, however, significant modifications on traditional DNS implementations. We analyse more in detail these two proposals in Section 4.5.

The proposals presented in [160, 161], as well as Tor, do not offer preservation of authenticity and integrity of DNS responses. Therefore, without other countermeasures, these solutions cannot avoid Man-in-the-Middle or replay attacks aiming at forging DNS responses. A proper solution for avoiding this problem is to combine the use of anonymity with the integrity and authenticity offered by the security extensions of DNS — often referred in the literature as DNSSEC (*cf.* Chapter 2, Section 2.2.1, for more information about DNSSEC). In this manner, we can guarantee the legitimacy of the response while maintaining an acceptable performance. We show in Section 4.6 that the impact on the latency of the service when using DNSSEC is minimal. We consider that authenticity and integrity threats are hence reduced by combining a proper anonymity model together with DNSSEC. None of these proposals guarantees the confidentiality of the queries. Although the use of alternative techniques such as IPSec [54] could be seen as a complementary solution to protect the exchanges on data between servers and clients of DNS, we consider that they are not appropriate for solving our motivation problem. First of all, the bandwidth and processing time overheads of using IPSec are much higher, and can render the solution impractical [97]. Secondly, IPsec does not offer protection during the caching processes between resolvers and/or intermediate servers. Furthermore, it is quite probable that servers of a global DNS service may not be IPsec capable. We consider that this approach is not an appropriate solution to our problem. Since our motivation is focused on privacy issues rather than confidentiality concerns, we consider that the combination of anonymity preservation together with integrity and authentication aspects offered by DNSSEC are worth enough to conduct our study.

4.4 Use of the Tor infrastructure to anonymise DNS queries

As we introduced in Chapter 2, multitude infrastructures aimed at reinforcing the anonymity of traffic directed to and through the Internet have been proposed in the literature. The main objective of these infrastructures is the concealment of the identity of its users. From simple proxies to complex cryptographic systems, these infrastructures help to reinforce both the anonymity of high-latency (*e.g.*, e-mail) and low-latency services (*e.g*, applications and Web services). One of the most commonly used infrastructures for navigating anonymously via the Web is the Tor (*The second generation Onion Router*), which is based on the use of a cryptographic scheme known as *onion routing*. The different components of the Tor project are currently distributed in open source mode and available for a large number of platforms and operating systems.

The maturity of the Tor project and its low impact on the performance of on-line services position it as an ideal candidate for our study on privacy in name resolution protocols. Even so, Tor clearly influences the performance of a critical service such as DNS. Motivated by the impact that Tor can have on the resolution of NAPTR-type queries introduced at the beginning of this chapter, we present in section 4.6 the results of a series of experiments aimed at analysing this penalty. We also analyse in this series of experiments the degree of anonymity that can be expected from the use of Tor. The whole set of tests was performed through Tor correctly, without experiencing serious problems or loss of messages. The disconnection of nodes in the Tor network caused, however, some fluctuations in the times analysed in our tests. During our experiments, measurements were made regarding the reliability of the nodes and circuits built in our scenarios. The result obtained is a reliability at the nodes of 88%, which leads to a reliability of 68% in each tunnel (assuming that the circuits are constructed with the default length

of three nodes). Given the nature of the DNS protocol, and the name resolution service that motivates the present work (NAPTR), we consider these results as acceptable.

To get such a low impact on traffic redirected through its nodes, Tor bases its security model on a really pragmatic scheme. First, Tor assumes the existence of active opponents in the network. These adversaries appear in the Tor model in order to compromise the identity of the users that send messages through their nodes. These opponents can not only observe, but also manipulate part of the redirected messages in Tor. A first implication of the model assumed by Tor is the complete access to the content of the messages that will transmit the exit nodes. In fact, if no countermeasures are applied, it is possible to carry out Man-in-the-Middle attacks on exit nodes. These attacks may involve, for example, the manipulation of responses of DNS queries performed through Tor. As a result, an adversary controlling exit nodes, and manipulating the responses of a DNS-based service, could redirect users to illegitimate servers, or deny the existence of a specific DNS record. As we introduced in Section 4.3, an efficient solution to this problem is the combination of Tor with the use of queries based on the DNS extensions proposed in DNSSEC. In this way, we can guarantee not only the authenticity and integrity of queries, but also the non existence of DNS records. As we show in Section 4.6, the impact on service latency through the combination of Tor and DNSSEC is minimal. Thus, we also consider that this first limitation in the Tor security model can be solved through DNSSEC-based queries. It is important to highlight that the same Tor infrastructure provides the DNS service for hostname resolution. When a user wants to solve a DNS query using Tor, the query is forwarded through an established circuit. When the query arrives to the last node of the circuit, the node is the responsible of resolving the query and send back the answer to the user. It is worth noting that the the current implementation does not allow the NAPTR queries and, at the same time, the resolution is not performed using the DNSSEC extension. In fact, the inclusion of DNSSEC in the hostname lookup is still a proposal pending to be implemented[1]. Consequently, the need of a custom development for this evaluation is justified.

A second implication of the security model associated with Tor is the possibility of having attacks based on traffic analysis. Again, the adversary's goal is to obtain the identity of the sender of messages passing through the nodes he controls. Several attacks of this type have been reported in the associated literature. The attack treated in [152, 153], often abbreviated as *predecessor-attack*, assumes that one or more adversaries control entry and exit nodes in many circuits of the Tor network. The cooperation between these nodes can be especially effective in degrading hidden services provided by the Tor network. Aside from providing anonymity to its users, Tor can provide anonymity for services that want to remain hidden behind the network. However, the possibility of confabulation between Tor nodes to cooperatively correlate information associated with these services (logins, for example) can significantly degrade Tor's anonymity. We consider that this is not the case of our study. We also think that the communications related to a DNS resolution are not easy to link as could be the correlation of Web traffic or SSH through the use of cookies or session identifiers. We do not therefore consider as relevant for our study other similar attacks described in [99, 108] and that are especially directed to the degradation of the hidden services of the Tor network.

An attack that has had a lot of impact among Tor users is the one presented in [100], where the authors propose to exploit the bandwidth limits offered by the nodes, and try to discover the nodes of

[1] https://gitweb.torproject.org/torspec.git/tree/proposals/219-expanded-dns.txt

the same circuit. This attack also works without the need of controlling Tor nodes, and is especially focused on discovering the entry node of circuits directed towards the same destination. The attack therefore assumes that an adversary must have complete control over the destination of the messages. The effectiveness of the proposed technique does not however seem to scale correctly with the current size of the Tor network. The authors propose in their work a set of improvements for the Tor network that increase the difficult to perform the attack. A more appropriate technique, and inspired by the previous attack, is the one presented in [17]. In this paper, the authors propose a new attack based on traffic analysis. They also argue that the attack can be executed even with a limited number of compromised nodes. Again, the attack proposes the cooperation between entry and exit nodes. It considers that the controlled nodes inject fraudulent information against the *Directory Servers* regarding their bandwidth and performance. In this way, when a Tor client requests information from the *Directory Servers* to build a new circuits, the controlled nodes will appear in privileged positions on the Tor node list. If this happens, the nodes controlled by the attacker will increase their chances of acting as entry and exit nodes in new circuits and, therefore, increase the probability of correlating information that allows the attacker to identify the sender of the messages. In our evaluation section (Section 4.6), we analyse in more detail the impact that the attack reported in [17] could cause to our test scenarios from the standpoint of the anonymity.

4.5 Use of random ranges to anonymise DNS queries

As an alternative to the use of anonymity infrastructures to increase the privacy of the DNS resolutions, we can consider the introduction of noise in the DNS queries. Although this proposal does not seem to have been widely studied, we can find some initial ideas presented by Zhao *et al.* in [160]. In fact, the model presented by the authors is inspired by the PIR (*Private Information Retrieval*) techniques [42, 107], used as a way to retrieve information from a database without revealing what information is wanted.

The approach presented by Zhao *et al.* works as follows: a user U, instead of launching just a single query to the DNS server NS, constructs a set of queries $Q\{H_i\}_{i=1}^{n}$. If we assume DNS queries of type A, the previous range of queries will include up to n different domain names to be resolved. The query $Q\{H_i\}$ will be the only one that includes the domain name desired by U. All the other queries in $Q\{H_1\} \dots Q\{H_{i-1}\}$ and $Q\{H_{i+1}\} \dots Q\{H_n\}$ are chosen at random from a database DB. The authors claim that this very simple model increases considerably the privacy of user U queries. Indeed, the only information disclosed by user U to third parties (*e.g.*, DNS server NS and possible attackers with either active or passive access to the channel between U and NS) is that the real query $Q\{H_i\}$ is within the interval $[1, n]$. Zhao *et al.* presume that the probability to successfully predict query $Q\{H_i\}$ requested by user U can be expressed as follows: $P_i = \frac{1}{n}$. We refer the reader to [160] for a more accurate description of the whole proposal.

However, we consider that the probability model presented in [160] is very optimistic. We believe that the degree of privacy offered by the model can clearly be degraded if we consider active attacks, in which an adversary is capable of interacting with the channel. Indeed, the approach does not address possible cases in which the resolution of query $Q\{H_i\}$ fails. In case of active attackers that can manipulate network traffic (*e.g.*, by means of RST attacks [10] or sending suitable ICMP traffic [131]),

they could launch a blind attack against the resolution protocol. This attack is based on dropping the query $Q\{H_i\}$ — or its associated response. Since attackers do not know which is the query-response pair desired by the client, they will try to force a fail resolution of every query $Q\{H_i\}_{i=1}^{n}$ and theirs associated responses. If so, user U will be forced to restart the process and generate a new range of queries — *i.e.*, requesting once again $Q\{H_i\}$. Depending on how this new range is managed, the degree of privacy estimated by the probabilistic model in [160] clearly decreases. Let $Q_j\{H_i\}_{i=1}^{n}$ be the j-th consecutive range exchanged for the resolution of the query $Q\{H_i\}$, the probability of success for an attacker trying to guess $Q\{H_i\}$ must then be defined as follows:

$$P_{ij} = \frac{1}{|Q_1\{H_i\}_{i=1}^{n} \cap Q_2\{H_i\}_{i=1}^{n} \cap \ldots \cap Q_j\{H_i\}_{i=1}^{n}|}$$

Let us exemplify this privacy level reduction attack by using the following ideal scenario. We assume a query range size of $n = 3$, a database of queries $\mathcal{DB}$= $\{H_1, H_2, H_3, H_4, H_5, H_6\}$, a DNS server $\mathcal{NS}$, and a client desired query resolution $Q\{H_1\}$. In the first stage of the protocol (*cf.* Table 4.2, Step 1), the client constructs a range query by choosing H_2 and H_3 from $\mathcal{DB}$ at random, resulting on $Q_1 = \{H_1, H_2, H_3\}$. Then, this range is sent to $\mathcal{NS}$ and intercepted by the attacker. In this step, from the point of view of the attacker, we can consider that the guess probability is $P_{i1} = 1/n = 1/3$. At this moment, we suppose that the attacker is able to lead a failed resolution of $Q\{H_1\}$ by manipulating the network traffic. Thus, the client is forced to construct (*cf.* Step 2) a new range $Q_2 = \{H_1, H_2, H_5\}$ which includes again H_1, and H_2 and H_5 are chosen randomly from $\mathcal{DB}$. When this new range is sent, the attacker can intercept it and calculate the intersection between the previous range and the current one, resulting on a privacy reduction, since $Q_1 \cap Q_2 = \{H_1, H_2\}$ and, consequently, $P_{i2} = 1/2$. Finally, we can see how, if the attacker successfully forces again an incomplete resolution of $Q\{H_1\}$ in Step 2, and intercepts the range $Q_3 = \{H_1, H_6, H_4\}$ built and sent by the client in Step 3, the attacker can deduce the desired query by simply applying the same intersection strategy among Q_2 and Q_3.

Zhao *et al.* present in [161] a second approach intended to reduce the bandwidth consumption imposed by the previous model. The new approach also gets inspiration from PIR approaches. It relies indeed on the construction of two ranges $Q_1\{H_i\}_{i=1}^{n}$ and $Q_2\{H_i\}_{i=1}^{n+1}$, where $H_{n+1} \in Q_2$ is the true query defined by user U. Once defined Q_1 and Q_2, such ranges are sent to two independent server NS_1 and NS_2. Assuming the resolution of DNS queries of type A, each server resolves every query associated with its range, obtaining all the associated IP addresses (defined in [161] as X_i) associated to the query H_i. NS_1 computes $R_1 = \sum_{i=1}^{n} \otimes X_i$ and NS_2 computes $R_2 = \sum_{i=1}^{n+1} \otimes X_i$. Both R_1 and R_2 are sent to user U, who obtains the resolution associated to H_{n+1} using the expression $X_{n+1} = R_1 \otimes R_2$. As we can observe, the bandwidth consumption of this new approach is considerably smaller than the one in [160], since only two responses (instead of n) are exchanged.

Step	Range	Intersection	Guess prob.
1	$Q_1 = \{H_1, H_2, H_3\}$	—	$P_{i1} = 1/3$
2	$Q_2 = \{H_1, H_2, H_5\}$	$Q_1 \cap Q_2 = \{H_1, H_2\}$	$P_{i2} = 1/2$
3	$Q_3 = \{H_1, H_6, H_4\}$	$Q_2 \cap Q_3 = \{H_1\}$	$P_{i3} = 1$

Table 4.2: Intersection attack against Zhao *et al.* protocol [160]

The main benefit of this last proposal, beyond the reduction of bandwidth consumption, is its achievement on preserving the privacy of the queries from attacks at the server side. However, it presents an important drawback due to the necessity of modifying DNS protocol and associated tools. Let us note that the proposal modifies the mechanisms for both querying the servers and responding to the clients. Moreover, it still presents security deficiencies that can be violated by means of active attacks against the communication channel between resolvers and servers. Indeed, attackers controlling the channel can still intercept both range Q_1 and Q_2. If so, they can easily obtain the true query established by user U by simply applying $Q_1 \setminus Q_2 = H_{n+1}$. Similarly, if attackers successfully intercept both R_1 and R_2 coming from servers NS_1 and NS_2, they can obtain the corresponding mapping address by performing the same computation expected to be used by user U, *i.e.*, by computing $X_{n+1} = R_1 \otimes R_2$. Once obtain such a value, they can simply infer the original query defined by user U by requesting a reverse DNS mapping of X_{n+1}. Analogously, an active control of the channel can lead attackers to forge resolutions. Indeed, without any additional measures, a legitimate user does not have non-existence proofs to corroborate query failures. This is especially relevant on UDP-based lookup services, like the DNS, where delivery of messages is not guaranteed. Attacker can satisfactorily apply these kind of attacks by intercepting, at least, one of the server responses. An attacker can for example intercept R_1, compute $R_2^* = R_1 \otimes R_3$ (where R_3 is a malicious resolution), and finally send as a resulting response coming from server NS_2. Then, the resolver associated to user U will resolve the mapping address as follows: $R_1 \otimes R_2^* = R_1 \otimes R_1 \otimes R_3 = R_3$.

As an alternative to the approaches presented in [160, 161], we propose to distribute the load of the set of ranges launched by user U among several servers $NS_1 \dots NS_m$. Unlike the previous schemes, our approach aims at constructing different ranges of queries for every server $NS_1 \dots NS_m$. The ranges will be distributed from $Q\{H_1^{NS_1}\} \dots Q\{H_{\frac{n}{m}}^{NS_1}\}$ to $Q\{H_1^{NS_m}\} \dots Q\{H_{\frac{n}{m}}^{NS_m}\}$. When the responses associated to these queries are obtained from the set of servers, user U verifies that the desired query has been successfully processed. If so, the rest of information is simply discarded. On the contrary, if the query is not processed, *i.e.*, user U does not receive the corresponding response, a new set of ranges is generated and proposed to the set of servers. To avoid the inference attack discussed above, ranges are constructed on independent sessions to preserve information leakage of the legitimate query. Let us note that by using this strategy, we preserve privacy of queries from both server and communication channel. In order to guarantee integrity of queries, authenticity of queries, and non-existence proofs, our proposal relies moreover on the use of the DNS security extension DNSSEC. The formal description of our proposed protocol is the following one:

- Let U be a user who wishes to perform anonymous resolution of a query $Q^*\{H\}$, and $\mathcal{DB}$ a database of queries.
- User U builds up a table Q of ranges, with every range $Q_j \in Q$ on the interval $j \in [1, m]$, and where the following properties apply:
  - $|Q_j| = n$ (the size of every range is n)
  - $\exists!\ v \in [1, m]$ such as $Q^*\{H\} \in Q_v$
  - $Q_j\{H_{ji}\}_{i=1}^n \neq Q^*\{H\}$ are selected at random from $\mathcal{DB}$, such that $\bigcap_{i=1}^n Q_j\{H_{ji}\} = \varnothing$

– $\bigcap_{j=1}^{n} Q_j = \varnothing$

- User U concurrently and randomly sends each range Q_j to a different server $\mathcal{NS}_w \;\; \forall w \in [1, m]$ with DNSSEC extensions enabled.
- User U verifies that all the responses have been properly received and their DNSSEC signatures are correct. Otherwise, the failed queries are retried until the responses are received and their signatures are correct, or until a certain number of retries R are achieved. In that case, the user is warned and the whole protocol is aborted.
- User U discards all those resolutions that are not associated to $Q^*\{H\}$.

4.6 Evaluation of the proposals

This section shows the outcome of our evaluation steered towards measuring the latency penalty due to the use of the approaches presented (the use of the Tor infrastructure and the use of random ranges of queries) on a real network scenario for the resolution of DNS and DNSSEC queries of type NAPTR. The hardware setup of our experimental scenario is the following. A host R, running on an Intel Core 2 Duo 2 GHz and 1 GB of memory, performs queries of type NAPTR to a global resolution service G. The global resolution service G is in turn implemented by means of three different hosts: S_1, that runs on an AMD Duron 1 GHz with 256 MB of memory; S_2, that runs on an Intel PIII 1 GHz with 512 MB of memory; and S_3, that runs on an Intel Xeon 2.4 GHz with 1 GB of memory. Servers in G are located on different networks and on different countries: server S_1 is located in North America; and servers S_2 and S_3 are located in Europe. DNS and DNSSEC services configured on each one of these hosts are based on BIND 9.4.2 (*cf.* `https://www.isc.org/downloads/bind/`). The configuration of each server in G consists of a database $\mathcal{N}$ that contains more than twenty thousand NAPTR records generated at random. Each one of these records are linked moreover with appropriate DNSSEC signatures. The different zones in $\mathcal{N}$ were signed by means of the *dnssec-keygen* and *dnssec-signzone* tools that come with BIND 9.4.2. The initial size of the database $\mathcal{N}$ is 6MB. The increase of the size of the database is 16MB after the inclusion of the digital signatures. Therefore, the final size of $\mathcal{N}$ is 22MB. The generation of the cryptographic keys was performed with *dnssec-keygen*, also included with the BIND 9.4.2. The key sizes are 1200 bits for the generation of Key Signing Keys (KSKs) and 1024 bits for Zone Signing Keys (ZSKs). The generation of keys is based on the cryptosystem RSA and the signatures on RSA/SHA1. Although the use of ECC signatures with DNSSEC seems to reduce the storage space of signed zones [2] compared to RSA or DSA, the algorithm we use is RSA instead of ECC since the latter is not yet implemented in BIND 9.4.2.

4.6.1 Evaluation of the model based on Tor

Four sets of tests are configured in this first evaluation in order to simulate the direct and indirect (through Tor) resolution of queries between R and G: (1) DNS resolutions; (2) DNSSEC resolutions; (3) DNS resolutions through Tor; and (4) DNSSEC resolutions through Tor. A direct connection is used in the first two sets of tests. We tag these tests as *Direct DNS tests* and *Direct DNSSEC tests*. An indirect connection based on SOCKS4a is used for the last two sets of tests. A Tor client version 0.1.2.18 (available at `https://www.torproject.org`) is executed in the system R redirecting the traffic

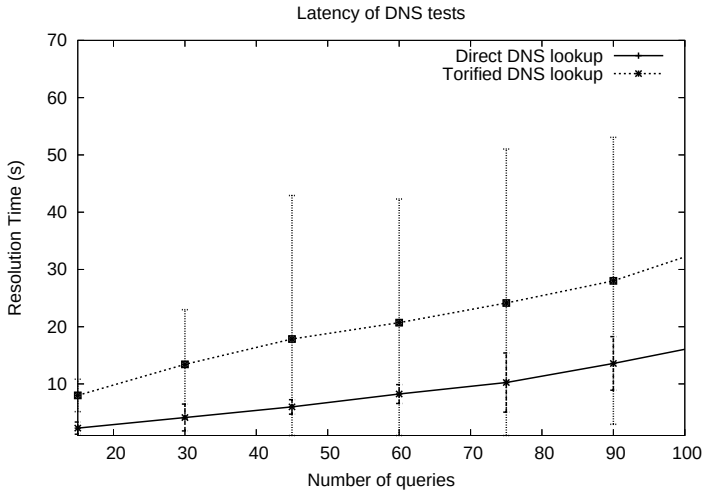
Bandwidth class					
996KB/s	621KB/s	111KB/s	59KB/s	29KB/s	<29KB/s
131	130	338	315	406	158

Table 4.3: Available nodes in the Tor network during the experiments, and classified by bandwidth

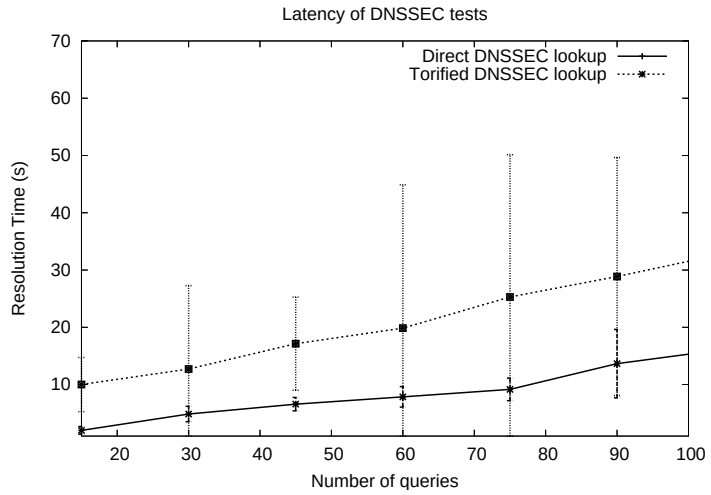
of the queries through messages SOCK4a towards the set of servers in G. We tag these two last tests as *Torified DNS tests* and *Torified DNSSEC tests*. The management of queries and responses DNS and DNSSEC in R are performed by means of an application based on the NET::DNS library (available at `https://www.net-dns.org`) and developed in *Perl*. Each query is executed as an independent process in R. The execution of n queries implies, therefore, the execution of n independent processes in R.

The activity and state of the Tor network at the start of our experiments is analysed by *TorFlow*, a set of scripts developed in Python and available at `https://svn.torproject.org/svn/torflow/`. The Table 4.3 shows a summary of the set of nodes available in the Tor network during our experiments, and classified according to the bandwidth reported by several *Directory Servers* (DS). We can observe that more than one thousand four hundred nodes are available for redirecting the traffic of our experiments. The Tor client installed in R is configured by default. For this reason, the length of the built circuits is three. The percentage of disconnection reported by *TorFlow* is 12% (with an error margin of 8%). According to [26], the reliability of the Tor circuits can be estimated as follows. Let l the length of the circuits (three nodes per circuit in our case). Let f the reliability of each node (88% as we have shown previously). The reliability of each circuit can be estimated as f^l. Thus, we can assume a 68% of reliability for each one of the Tor circuits that are used in our experiments.

We show in Figure 4.1 the results that we obtained during the execution of these four experiments. Figure 4.1a shows the execution of the tests tagged as *Direct DNS tests* and *Torified DNS tests*. Figure 4.1b depicts the results of the tests tagged as *Direct DNSSEC tests* and *Torified DNSSEC tests*. Each group of tests is executed several times in order to generate different series of random queries from the set $\mathcal{N}$. Each series is stored persistently during the execution of the firs set of tests (*Direct DNS tests*) and loaded in the rest of tests — with the aim of facilitating the comparison of results. On the first hand, we can appreciate by looking at the first curve of Figure 4.1a and 4.1b some minor differences between the queries based on DNS and DNSSEC. On the other hand, each series of tests that belongs to the *Torified DNS* and *Torified DNSSEC* experiments is executed using different Tor circuits. As we described in Section 4.4, the dynamic disconnections of nodes in the Tor network lead the creation of more circuits for each series. Consequently, the application in R was forced to repeat the queries in some cases. These node disconnections are reflected in Figure 4.1a and 4.1b with the worst times shown in the confidence intervals for each series. In spite of this, we would like to remark the fact that under such extreme cases the total times can be considered as acceptable. Moreover, all the queries were evaluated and resolved. We also consider that the impact of including DNSSEC with Tor is negligible, acceptable, and recommended, specially, if we take into account that it provides security properties such as integrity, authenticity and denial-of-existence. These properties are essential to detect and prevent Man-In-the-Middle attacks that could be perpetrated by Tor exit nodes. In that sense, we did not experience any alteration of the signatures related to the response records in $\mathcal{N}$, and queried against G through the Tor



(a) Direct vs. torified DNS tests



(b) Direct vs. torified DNSSEC tests

Figure 4.1: Experimental results of the evaluation of the Tor model

network.

Motivated by the need of knowing the degree of anonymity that we should expect during the experiments, we adopted the same strategy presented in [17]. We estimated the anonymity degree by means of the probability distribution associated to the Tor nodes [50, 122], and that we will formalise in Chapter 5. Let N the total number of nodes in the Tor network. Let $p(x_i)$ the probability that the node x_i is selected to be part of a circuit. We can calculate the degree of anonymity by means of the following metric based on the concept of entropy [125]:

$$H(N) = -\sum_{x_i \in N} p(x_i) \cdot log_2(p(x_i))$$

According to the previous expression, if each Tor node has the same probability of being included in a circuit, and the measured entropy $H(N)$ is normalised by dividing by $log_2(|N|)$, we will obtain the

maximum degree of anonymity 1. However, the algorithm for the construction of Tor circuits associates a higher probability to the nodes that exhibit better performance in terms of bandwidths, or up-times among others. Thus, we can not assume that the probability will be the same for each node. Using the tool *TorFlow*, we approximate the value $H(N)$ by grouping the general set of nodes in different categories according to their bandwidths, and by building several circuits associated to each category. The estimated normalised entropy obtained is 0.89.

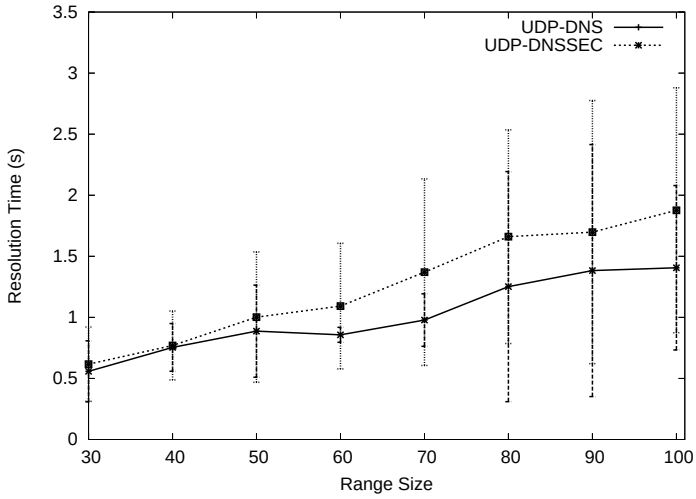
As we introduced in Section 4.4, the security model of Tor presents certain vulnerabilities that could be exploited by an attacker with the aim of degrading the previous value. If the attacker controls an elevated number of nodes in the network, it can try to correlate information reported by entry and exit nodes that belong to the same circuit. Under such circumstances, the attacker could obtain the location (IP address) of the client that built the circuit, the destination (IP address) of the messages, and potentially the content of each message. The main goal of the attacker is then to guarantee that their nodes have a higher probability of being selected during the construction of the circuits. In accordance with the original developers of Tor [52], if an adversary controls $m > 1$ nodes from a total of N, it could potentially correlate the traffic of the network with a probability of $\left(\frac{m}{N}\right)^2$. In [17], the authors show that by injecting false information in the *Directory Servers* of Tor, it is possible to increment the selection probability associated to the nodes. In fact, the authors claim that the previous model should be replaced by $\left(\frac{m}{n}\right)\left(\frac{m-1}{N-1}\right)$, arguing that this new model takes into account that a node is only used one time per circuit.

Using this second model, the authors present an experimental implementation of their proposal using a private Tor network deployed under PlanetLab [113]. The results that they obtained turned out to be seventy times better that the expected analytical values, being the number of controlled nodes from 5% to 10% of the total of the network. The same percentage of compromised nodes in the network used four our experiments would suppose that a hypothetical attacker controls from seventy to more than one hundred nodes. The analytical prediction from the proposed model would indicate that the number of potentially compromised circuits in this case would be between 0.21% and 0.67% of the total circuits of the Tor network. Assuming that the attack presented in [17] scales correctly in the network, and maintaining the improvement reported by the authors, we should consider that the degree of anonymity obtained from the network during our experiments would be, in the best case, close to a value $H(N)$ of 0.89 and, in the worst case assuming an attack like the one presented in [17], around 0.45.

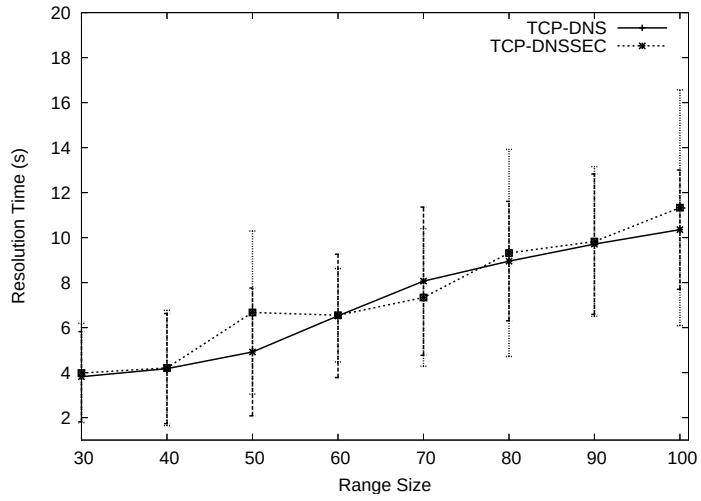
4.6.2 Evaluation of the model based on PIR

The implementation and deployment of our proposal in R for the evaluation of the PIR model is developed in Python language. More specifically, we base our implementation on the module *dnspython* [105] for the construction and resolution of DNS queries; and the module *M2Crypto* [132] (a wrapper for the OpenSSL library [143]) for the verification of digital signatures defined by DNSSEC.

We measured in our evaluations the time required for resolving queries from R to G with different testbeds, where the size of the query range of each testbed increments from thirty to more than one hundred. Each testbed consists indeed on the generation of three sets of random queries, one for each $S_i \in G$. Each testbed is launched multiple times towards cumulative series of NAPTR queries. Each



(a) UDP tests



(b) TCP tests

Figure 4.2: Experimental results of the evaluation of the PIR model

series is created at random during the execution of the first testbed, but persistently stored. It is then loaded into the rest of testbeds to allow comparison of results. We split our whole evaluation in four different stages. During the first two stages, the transport layer utilised between R and G is based on the TCP protocol. First stage is used for the resolution of DNS queries, while stage two is used to resolve DNSSEC queries. Similarly, stage three and four are based on UDP traffic for the resolution of, respectively, DNS and DNSSEC queries. During these two last experiments based on DNSSEC, R verifies the integrity and the authenticity of the queries received from the different servers in G. The verification procedures have been implemented as defined in DNSSEC RFCs (*cf.* Chapter 2, Section 2.2.1). We show in Figure 4.2 the results that we obtained during the execution of these four experiments.

We can appreciate by looking at Figure 4.2 that the latency increases linearly with the size of the range of queries. TCP-based experiments show worst performance than UDP-based queries — due to

the overhead imposed by the establishment of sessions. UDP protocol is clearly the best choice for the deployment of our proposal. Given an acceptable latency of no more than two seconds, UDP results show that the probability of guessing the true query is $P_i = \frac{1}{3 \cdot 80} = \frac{1}{240} \simeq 0.004167$. We consider this result as satisfactory. In general terms, we should expect that the certainty for obtaining a query i within a range of size n and m different servers is $P_i = \frac{1}{n \cdot m}$.

Besides the difficulties imposed by our model for predicting the original petition, we are conscious of the high bandwidth increase that it represents. This is an important drawback in scenarios where the bandwidth consumption is a critical factor. However, if this is the case, it is possible to reduce the size of the range of queries. Since there is a clear relation between both parameters, *i.e.*, the bandwidth consumption is inversely proportional to the prediction probability, we believe that a proper balance between bandwidth consumption and prediction probability can be enough to enhance the privacy of the service. Let us recall that reducing the size of each range of queries to a fifty per cent, the prediction probability for the attacker is proportionally increased by two. On the other hand, let us observe how the penalty in the response times introduced by DNSSEC is not specially significant, solving the integrity and authenticity problems that appeared in the other approaches. This is the reason why we consider the activation of DNSSEC as a decisive factor for avoiding manipulation network traffic attacks.

4.7 Conclusion

The use of the DNS as the underlying technology of new lookup services, such the ENUM protocol, might have unwanted consequences from the point of view of security and privacy. We have analysed two proposal that could mitigate the privacy problems aforementioned: the use of the Tor infrastructure, and the use of range of queries.

On the one hand, we have analysed the network latency of Tor performing NAPTR DNS queries, as well as the degree of anonymity. Taking into account the security model of Tor, we consider the results obtained as very satisfactory. In addition, and in order to guarantee the integrity and authenticity of the received responses, we have also analysed the implication of combining the anonymity offered by Tor together with the use of DNSSEC. The results obtained are also satisfactory with a minimum penalty.

On the other hand, we have implemented an approach inspired on a PIR model. The goal of our model is to reduce privacy threats at both channel and server level. The proposal is indeed inspired on two previous works surveyed by Zhao *et al.* Security deficiencies detected in both contributions have been addressed. Again, the combination of our model with the use of DNSSEC has had a minimal impact. The main drawback of this contribution is still a high increase on the bandwidth consumption of the service.

5

Formal modelling of Tor node selection criteria

> *The world we have made, as a result of the level of thinking we have done thus far, creates problems we cannot solve at the same level of thinking at which we created them.*
>
> ALBERT EINSTEIN

As we have introduced previously, Tor allows the construction of anonymous channels with latency enough to route traffic for services like the DNS. However, it might still impact its network performance and degree of anonymity depending on the specific strategy used for the establishment of the channel. In this chapter, we address the influence of circuit construction strategies on the anonymity degree of Tor. In particular, we introduce a formal model providing a definition of the selection of Tor nodes process, of the adversary model targeting the communication anonymity of Tor users, and an analytical expression to compute the anonymity degree of the Tor infrastructure based on the circuit construction criteria. This formal model becomes an useful tool as a way to compare different node selection algorithms from the standpoint of the degree of anonymity. In conjunction with network latency measurements, it can allow a user to choose a particular selection algorithm depending on its needs and regarding the trade-off between degree of anonymity and network performance. We also show how this formal model can allow to infer other underlying properties of the algorithms.

5.1 Formal model

In this section, we introduce our formal model composed by four related topics: *the Tor circuit*, *the adversary model*, *the degree of anonymity*, and *the selection criteria*. Following, we introduce the notation and core definitions for each one of them.

5.1.1 Tor circuit

Formally, we can describe a connection using the Tor network as follows. First, we define a client node s called a *client* or *onion proxy*, and a *destination server* node d which we want to interconnect to exchange data in an anonymous manner. Let N be the set of nodes deployed in the Tor network, and $n = |N|$ the cardinality of the set. Let node $e \in N$ denote a specified node, called the *entrance node*, and $x \in N$ the *exit node*. Then, a *Tor circuit* is a sequence of nodes $C = \langle s, e, r_1, r_2, ..., r_l, x \rangle$, where $r_i \in N$ is any *intermediary node*. The nodes e, x, and r_i, $i \in \{1, ..., l\}$, are also known as *onion routers*. We define the *path of a circuit* as the set of links (*i.e.*, network connections) $P = \{a_1, ..., a_{l+2}\}$ associated to the *Tor circuit*, where $a_1 = (s, e)$, $a_2 = (e, r_1)$, $a_3 = (r_1, r_2)$, ... , $a_{l+1} = (r_{l-1}, r_l)$, $a_{l+2} = (r_l, x)$. The value $|P| = l + 2$ is called the *length of the circuit*. A *connection using the Tor network* is composed by the client and destination nodes interconnected through a Tor circuit as follows:

$$\underbrace{s \xrightarrow{a_1} e \xrightarrow{a_2} r_1 \xrightarrow{a_3} r_2 \xrightarrow{a_4} ... \xrightarrow{a_l} r_{l-1} \xrightarrow{a_{l+1}} r_l \xrightarrow{a_{l+2}} x}_{\text{Tor network}} \rightarrow d$$

5.1.2 Adversary model

The adversary assumed in our work relies on the threat model proposed by Syverson *et al.* in [140]. Such a pragmatic model considers that, regardless of the number of onion routers in a circuit, an adversary controlling the entrance and exit nodes would have enough information in order to compromise the communication anonymity of a Tor client. Indeed, when both nodes collude, and given that the entry node knows the source of the circuit, and the exit node knows the destination, they can use traffic analysis to link communication over the same circuit [74].

Assuming the model proposed in [140], then an adversary who controls $c > 1$ nodes over the n nodes in the Tor network can control an entry node with probability $(\frac{c}{n})$, and an exit node with probability $(\frac{c}{n})$. This way, the adversary may de-anonymise the traffic flowing on a controlled circuit (*i.e.*, a circuit whose entry and exit nodes are controlled by the adversary) with probability $(\frac{c}{n})^2$ if the length of the circuit is greater than two; or $\frac{c(c-1)}{n^2}$ if the length of the circuit is equal to two (*cf.* [140] and citations thereof). Adversaries can determine when the nodes under their control are either entry or exit nodes for the same circuit stream by using attacks such timing-based attacks [13], fingerprinting [99], and several other existing strategies.

Let us observe that the aforementioned probability of success assumes that the probability of a node from being selected on a Tor circuit is randomly uniform, that is, the boundaries provided in [140] only apply to the standard (random) selection of nodes, hereinafter denoted as *random selection of nodes strategy*. Given that the goal of our research work in this chapter is to evaluate alternative selection strategies, we shall adapt the model. Therefore, let p_1, p_2, p_3, ..., p_c be the corresponding selection

probabilities assigned by the circuit construction algorithm to each node controlled by the adversary, then the probability of success corresponds to the following expression:

$$(p_1 + p_2 + p_3 + \ldots + p_c) \cdot (p_1 + p_2 + p_3 + \ldots + p_c)$$

that can be simplified as:

$$\Big(\sum_{i=1}^{c} p_i\Big)^2$$

Following is the analysis.

Theorem 1. *Let c be the number of nodes controlled by the adversary. Let the Tor client use a selection criterion which, for a certain circuit, every node selection is independent. Let p_1, p_2, p_3, ..., p_c be the corresponding selection probabilities assigned by the circuit construction algorithm to each node controlled by the adversary. Then, the success of the adversary to compromise the security of the circuit is bounded by the following probability:*

$$\Big(\sum_{i=1}^{c} p_i\Big)^2$$

Proof. The proof is direct by using the sum and product rules of probability theory, and taking into account that the selection of every node is an independent event. First, the probability of selecting the entrance or exit node in the set of nodes controlled by the adversary is (sum rule):

$$\sum_{i=1}^{c} p_i$$

Then, the probability of selecting, at the same time, a controlled entrance and exit node in a circuit is (product rule):

$$\Big(\sum_{i=1}^{c} p_i\Big)\Big(\sum_{i=1}^{c} p_i\Big) = \Big(\sum_{i=1}^{c} p_i\Big)^2$$

□

Corollary 2. *The Syverson et al. success probability boundary in [140], i.e., $(\frac{c}{n})^2$, is equivalent to the boundary defined in Theorem 1 when the circuit selection criterion is a random selection of nodes.*

Proof. Let N be the set of nodes deployed in a Tor network with $n = |N|$, and let $A \subseteq N$ be the subset of nodes controlled by an adversary with $c = |A|$. The probability of a node $n_i \in N$ to be selected is $p_i = \frac{1}{n}$. Then, by applying it to the boundary defined in Theorem 1, we obtain:

$$\Big(\sum_{i=1}^{c} p_i\Big)^2 = (c \cdot p_i)^2 = \Big(c\frac{1}{n}\Big)^2 = \Big(\frac{c}{n}\Big)^2$$

□

5.1.3 Anonymity degree

Most work in the related literature has used the Shannon entropy [125] concept to measure the anonymity degree of anonymisers like Tor (*cf.* [50, 122] and citations thereof). We recall that the

entropy is a measure of the uncertainty associated with a random variable, that can efficiently be adapted to address new security research problems [3, 11, 84]. In this chapter, the entropy concept is used to determine how predictable is the selection of the nodes in accordance to a given strategy or, in other words, how easy is to put in risk the anonymity in relation to the adversary model defined in Section 5.1.2. This is possible since the selection criteria of nodes for the construction of a circuit can be modelled as a random variable, where the choice of every node has a particular probability. Thus, the Shannon entropy is useful since it provides a way to measure the uncertainty contained in such probability distribution.

Formally, given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with a sample space $\Omega = \{\omega_1, \omega_2, ..., \omega_n\}$ where ω_i denotes the outcome of the node $n_i \in N$ ($\forall i \in \{1, ..., n\}$), a σ-field $\mathcal{F}$ of subsets of Ω, and a probability measure $\mathbb{P}$ on $(\Omega, \mathcal{F})$, we consider a random discrete variable X defined as $X : \Omega \to \mathbb{R}$ that takes values in the countable set $\{x_1, x_2, ..., x_n\}$, where every value $x_i \in \mathbb{R}$ corresponds to the node $n_i \in N$. The discrete random variable X has a pmf (probability mass function) $f : \mathbb{R} \to [0, 1]$ given by $f(x_i) = p_i = \mathbb{P}(X = x_i)$. Then, we define the entropy of a discrete random variable (*i.e.*, the entropy of a Tor network) as:

$$H(X) = -\sum_{i=1}^{n} p_i \cdot log_2(p_i) \tag{5.1}$$

Since the entropy is a function whose image depends on the number of nodes, with property $H(X) \geq 0$, it cannot be used to compare the level of anonymity of different systems. A way to avoid this problem is as follows. Let $H_M(X)$ be the maximal entropy of a system, then the entropy that the adversary may obtain after the observation of the system is characterised by $H_M(X) - H(X)$. The maximal entropy $H_M(X)$ of the network applies when there is a uniform distribution of probabilities (*i.e.*, $\mathbb{P}(X = x_i) = p_i = \frac{1}{n}$, $\forall i \in \{1, ..., n\}$), and this leads to $H(X) = H_M(X) = log_2(n)$. The anonymity degree shall be then be defined as:

$$d = 1 - \frac{H_M(X) - H(X)}{H_M(X)} = \frac{H(X)}{H_M(X)} \tag{5.2}$$

Note that by dividing $H_M(X) - H(X)$ by $H_M(X)$, the resulting expression is normalised. Therefore, it follows immediately that $0 \leq d \leq 1$.

5.1.4 Selection criteria

Taking into account the aforementioned anonymity degree expression, we can now formally define a selection of Tor nodes criterion as follows.

Definition 26. *A selection of Tor nodes criterion is an algorithm executed by a Tor client s that, from a set of nodes N with $n = |N|$ and a length of a circuit δ, selects —using a given policy— the entrance node e, the exit node x, and the intermediary nodes r_i, $\forall i \in \{1, ..., \delta - 2\}$, and outputs its corresponding circuit $C = \langle s, e, r_1, r_2, ..., r_{\delta-2}, x\rangle$ with a path $P = \{a_1, ..., a_\delta\}$, where $a_1 = (s, e)$, $a_2 = (e, r_1)$, $a_3 = (r_1, r_2)$, ... , $a_{\delta-1} = (r_{\delta-3}, r_{\delta-2})$, $a_\delta = (r_{\delta-2}, x)$. We use the notation convention $\psi(N, \delta)$ to denote the algorithm. The policy for the selection criterion of nodes can be modelled as a discrete random variable X that has a pmf $f(x)$, and we use the notation $\psi(N, \delta) \sim f(x)$.*

Algorithm 5.1 Random Selection of Nodes - $\psi_{rnd}(N, \delta)$

Input: s, N, δ
Output: $C = \langle s, e, r_1, r_2, ..., r_{\delta-2}, x\rangle$, $P = \{a1, ..., a_\delta\}$

$M := N$
$C := \{s\}$
for $i := 1$ **to** δ **do**
    $j :=$random$(1, |M|)$
    $C := C \cup \{m_j \mid m_j \in M\}$
    $P := P \cup \{(c_i, c_{i+1})\}$
    $M := M \setminus \{m_j \mid m_j \in M\}$
end for

5.2 Anonymity degree of three classical circuit construction strategies

In this section, we present three existing strategies for the construction of Tor circuits, and elaborate on the conceptual evaluation of their anonymity degree. This will illustrate to the reader how the previous presented formal model can be applied to some particular selection of nodes strategies.

5.2.1 Random selection of nodes

The random selection of Tor nodes is an algorithm $\psi_{rnd}(N, \delta) \sim f_{rnd}(x)$ with an associated discrete random variable X_{rnd}. The procedure associated to this selection criterion is outlined in Algorithm 5.1. The selection policy of $\psi_{rnd}(N, \delta)$ is based on uniformly choosing at random those nodes that will be part of the resulting circuit. Thus, the pmf $f_{rnd}(x)$ is defined as follows:

$$f_{rnd}(x_i) = p_i = \mathbb{P}(X_{rnd} = x_i) = \frac{1}{n}$$

Hence, the entropy of a Tor network whose clients use a random selection of nodes is characterised by the following expression:

$$\begin{aligned} H_{rnd}(X_{rnd}) &= -\sum_{i=1}^{n} \frac{1}{n} \cdot log_2\Big(\frac{1}{n}\Big) = \\ &= -\frac{1}{n}\sum_{i=1}^{n} (log_2(1) - log_2(n)) = \\ &= log_2(n) \end{aligned}$$

Theorem 3. *The selection of Tor nodes $\psi_{rnd}(N, \delta) \sim f_{rnd}(x)$ with an associated discrete random variable X_{rnd} gives the maximum degree of anonymity among all the possible selection algorithms.*

Proof. The proof is direct by replacing $H_{rnd}(X_{rnd})$ in Equation (5.2):

$$d_{rnd} = \frac{H_{rnd}(X_{rnd})}{H_M(X_{rnd})} = \frac{log_2(n)}{log_2(n)} = 1$$

□

Algorithm 5.2 Geographical Selection of Nodes - $\psi_{geo}(N, \delta)$

Input: s, N, δ, K_c
Output: $C = \langle s, e, r_1, r_2, ..., r_{\delta-2}, x\rangle$, $P = \{a1, ..., a_\delta\}$

$M := \{n_i \in N \mid g_c(n_i) = K_c\}$
$C := \{s\}$
for $i := 1$ **to** δ **do**
    $j :=$random$(1, |M|)$
    $C := C \cup \{m_j \mid m_j \in M\}$
    $P := P \cup \{(c_i, c_{i+1})\}$
    $M := M \setminus \{m_j \mid m_j \in M\}$
end for

5.2.2 Geographical selection of nodes

The geographical selection of Tor nodes is an algorithm $\psi_{geo}(N, \delta) \sim f_{geo}(x)$ with an associated discrete random variable X_{geo}. Its selection method is based on uniformly choosing the nodes that belong to the same country of the client s that executes $\psi_{geo}(N, \delta)$. The aim of this strategy is to reduce the latency of the communications using the Tor network, since the number of hops between Tor nodes of the same country is normally smaller than the number of hops between nodes that are located at different countries. Algorithm 5.2 summarises the procedure of this selection criterion.

Formally, we define a function $g_c : \mathbb{R} \to \mathbb{N}$ that, given a certain node $x_i \in X_{geo}$, returns a number that identifies its country. Thus, given the specific country number K_c of the client node s, the pmf $f_{geo}(x)$ is characterised by the following expression:

$$f_{geo}(x_i) = p_i = \mathbb{P}(X = x_i) = \begin{cases} \frac{1}{m}, & \text{if } g_c(x_i) = K_c; \\ 0 & \text{otherwise.} \end{cases}$$

where $m = |\{x_i \in X_{geo} \mid g_c(x_i) = K_c\}|$. Then, the entropy of a system whose client nodes use a geographical selection for a certain country K_c is:

$$H_{geo}(X_{geo}) = -\sum_{i=1}^{m} \frac{1}{m} \cdot log_2\Big(\frac{1}{m}\Big) = log_2(m)$$

Therefore, by replacing the previous expression in Equation (5.2), the anonymity degree is equal to:

$$d_{geo} = \frac{log_2(m)}{log_2(n)}$$

Theorem 4. *The maximum anonymity degree of a Tor network whose clients use a geographical selection of nodes is achieved iff all the nodes are in the same fixed country K_c.*

Proof. ($\Rightarrow$) Given $d_{geo} = \frac{log_2(m)}{log_2(n)}$ for the country K_c of a particular client s, we can impose the restriction of maximum degree of anonymity:

$$d_{geo} = \frac{log_2(m)}{log_2(n)} = 1$$

Hence,

$$log_2(m) = log_2(n)$$
$$2^{log_2(m)} = 2^{log_2(n)}$$
$$m = n$$

($\Leftarrow$) If $g_c(x_i) = K_c$, $\forall x_i \in X_{geo}$, then we have that $m = |\{x_i \in X_{geo} \mid g_c(x_i) = K_c\}| = |N|$. Thus,

$$d_{geo} = \frac{log_2(m)}{log_2(n)} = \frac{log_2(n)}{log_2(n)} = 1$$

□

Theorem 5. *Given a Tor network whose clients use the algorithm $\psi_{geo}(N, \delta) \sim f_{geo}(x)$ for a fixed country K_c, and with an associated discrete random variable X_{geo}, the anonymity degree is increased as m approaches n (i.e., $m \to n$), where $m = |\{x_i \in X_{geo} \mid g_c(x_i) = K_c\}|$ and $n = |N|$.*

Proof. It suffices to prove that d_{geo} is a monotonically increasing function. That is, we must prove that $\frac{\partial}{\partial m}(d_{geo}) > 0$, $\forall m > 0$. Therefore, the proof is straightforward, since the inequality:

$$\frac{\partial}{\partial m}\Big(\frac{log_2(m)}{log_2(n)}\Big) = \frac{1}{m \cdot log(n)} > 0$$

is true $\forall m > 0$ and $\forall n > 1$. We must notice that, from the point of view of a Tor network, the restriction of the number of nodes $n > 1$ makes sense, since a network with $n \leq 1$ nodes becomes useless as a way to provide an anonymous infrastructure. □

Figure 5.1 depicts the influence of the uniformity of the number of nodes per country on the anonymity degree. It shows, for a fixed country, the anonymity degree of four Tor networks in function of the nodes that are located in that country with respect to the total number of nodes of the network. The considered Tor networks have, respectively, 10, 50, 100 and 200 nodes. Their anonymity degrees are denoted as d_{10}, d_{50}, d_{100} and d_{200}. We can observe that the anonymity degree increases as the total number of nodes of the same country grows up (*cf.* Theorem 5). This fact can be extended until the maximum value of anonymity is achieved, which occurs when the number of nodes of the particular country is the same as the nodes that compose the entire network (*cf.* Theorem 4).

Theorem 6. *Given a client s that uses as selection algorithm $\psi_{geo}(N, \delta)$ in a Tor network with $n = |N|$, such that the network nodes belong to a $p \ll n$ different countries, where p is the number of different countries in Tor network, then the best distribution of nodes that maximises the anonymity degree of the whole system is achieved iff every country has $t = \lfloor \frac{n}{p} \rceil$ nodes.*

Proof. ($\Rightarrow$) Let p be the number of different countries of a Tor network, we can consider a collection of subsets $S_1, S_2, ..., S_p \subseteq N$ such as $\bigcup_{i=1}^{p} S_i = N$ and $\bigcap_{i=1}^{p} S_i = \varnothing$. Let t_i be the number of nodes associated to the subset S_i, $i \in \{1, ..., p\}$. Then, the anonymity degree of the whole system is maximised

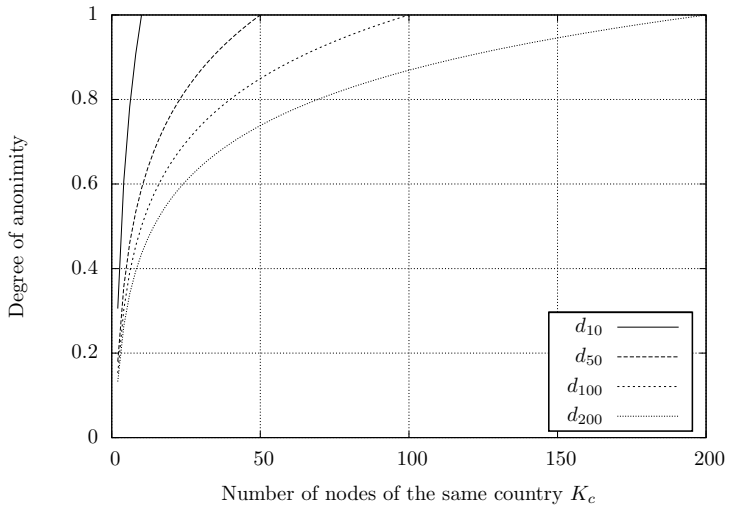


Figure 5.1: Influence of the uniformity of the number of nodes per country in the anonymity degree for $\psi_{geo}(N, \delta)$

when the sum of all the degrees of anonymity of every country equals 1:

$$\sum_{i=1}^{p} \frac{log_2(t_i)}{log_2(n)} = 1$$

$$\frac{log_2(t_1)}{log_2(n)} + \frac{log_2(t_2)}{log_2(n)} + ... + \frac{log_2(t_p)}{log_2(n)} = 1$$

$$2^{log_2(t_1)} + 2^{log_2(t_2)} + ... + 2^{log_2(t_p)} = 2^{log_2(n)}$$

$$t_1 + t_2 + ... + t_p = n$$

However, to maximise the anonymity degree of the whole system implies also to have the same uncertainty inside every subset S_i, $i \in \{1, ..., p\}$, or, in other words, to have the same number of nodes in every subset. Hence, we have $t_1 = t_2 = ... = t_p = t$ and this leads to:

$$t_1 + t_2 + ... + t_p = n$$

$$\underbrace{t + t + ... + t}_{p \text{ times}} = n$$

$$p \cdot t = n$$

$$t = \frac{n}{p}$$

($\Leftarrow$) Given $t = \lfloor \frac{n}{p} \rceil$ be the number of nodes of a certain subset S_i, $i \in \{1, ..., p\}$, we have $\sum_{i=1}^{p} |S_i| = p \cdot t = n$. The pmf associated to $\psi_{geo}(N, \delta)$ is then $f_{geo}(x) = \frac{1}{t}$ for each subset S_i, $i \in \{1, ..., p\}$. Therefore, the entropy of each subset (*i.e.*, country) is:

$$H_{geo}(X_{geo}) = -\sum_{i=1}^{t} \frac{1}{t} \cdot log_2\Big(\frac{1}{t}\Big) = log_2(t)$$

Hence, for each subset S_i, $i \in \{1, ..., p\}$, the anonymity degree can be expressed as follows:

$$d_{geo} = \frac{log_2(t)}{log_2(n)}$$

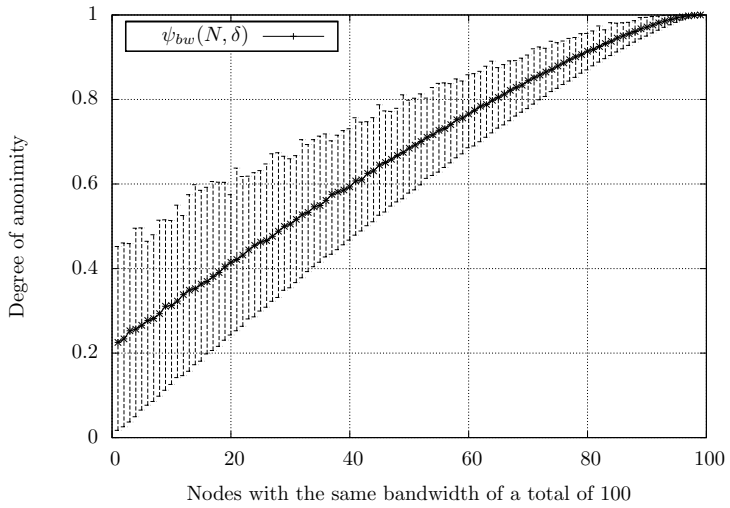


Figure 5.2: Influence of the uniformity of the bandwidth distribution in the anonymity degree for $\psi_{bw}(N,\delta)$

Suppose now, by contradiction, that there exists a unique $S_q \in \{S_1, S_2, ..., S_p\}$ for a particular country K_q such that $|S_q| \neq t$, and its anonymity degree is expressed by $d_{geo^*} = \frac{log_2(|S_q|)}{log_2(n)}$. Then, taking into account that d_{geo} and d_{geo^*} are monotonically increasing functions (*cf.* proof of Theorem 5), we have two options:

- If $|S_q| < t \rightarrow d_{geo^*} < d_{geo}$
- If $|S_q| > t \rightarrow d_{geo^*} > d_{geo}$

But this is not possible since:

$$\begin{aligned} \sum_{i=1}^{p} |S_i| &= n \\ (p-1)t + |S_q| &= n \\ |S_q| &= n - t(p-1) \\ |S_q| &= n - \frac{n}{p}(p-1) \\ |S_q| &= \frac{n}{p} \end{aligned}$$

which implies that $d_{geo^*} = d_{geo}$, contradicting the above two options. □

5.2.3 Bandwidth selection of nodes

The bandwidth selection of nodes strategy is an algorithm $\psi_{bw}(N,\delta) \sim f_{bw}(x)$ with an associated discrete random variable X_{bw} whose selection policy is based on choosing, with high probability, the nodes with best network bandwidth. The aim of this strategy is to reduce the latency of the communications through a Tor circuit, specially when the communications imply a great rate of data exchanges. At the same time, this mechanism provides a balanced anonymity degree, since the selection of nodes is not fully deterministic from the adversary point of view.

In this strategy, the entropy and the anonymity degree can be described formally as follows. First, we define a bandwidth function $g_{bw} : \mathbb{R} \to \mathbb{N}$ that, given a certain node $x_i \in X_{bw}$, returns its associated bandwidth. Then, the pmf $f_{bw}(x)$ is defined by the expression:

$$f_{bw}(x_i) = p_i = \mathbb{P}(X_{bw} = x_i) = \frac{g_{bw}(x_i)}{T_{bw}}$$

where $T_{bw} = \sum_{i=1}^{n} g_{bw}(x_i)$ is the total bandwidth of the Tor network. Hence, the entropy of a system whose clients use a bandwidth selection of nodes strategy is:

$$H_{bw}(X) = -\sum_{i=1}^{n} \frac{g_{bw}(x_i)}{T_{bw}} \cdot log_2\left(\frac{g_{bw}(x_i)}{T_{bw}}\right)$$

By replacing $H_{bw}(X)$ in Equation (5.2), the anonymity degree is, then, as follows:

$$d_{bw} = -\sum_{i=1}^{n} \frac{g_{bw}(x_i)}{T_{bw} \cdot log_2(n)} \cdot log_2\left(\frac{g_{bw}(x_i)}{T_{bw}}\right)$$

Theorem 7. *Given a selection of Tor nodes $\psi_{bw}(N, \delta) \sim f_{bw}(x)$ with an associated discrete random variable X_{bw}, the maximum anonymity degree is achieved iff $g_{bw}(x_i) = K_{bw}\ \forall x_i \in X_{bw}$, where K_{bw} is a constant.*

Proof. ($\Rightarrow$) $H(X_{bw}) = H_M(X_{bw})$ would imply that the anonymity degree gets maximum. This is only possible when $f_{bw}(x_i) = \frac{g_{bw}(x_i)}{T_{bw}} = \frac{1}{n}$, $\forall x_i \in X_{bw}$. Therefore,

$$\frac{g_{bw}(x_i)}{T_{bw}} = \frac{1}{n}$$
$$g_{bw}(x_i) = \frac{T_{bw}}{n}$$

and since T_{bw} and n are constant values for a certain Tor network, we can consider that $g_{bw}(x_i)$ is also a constant, $\forall x_i \in X_{bw}$.

($\Leftarrow$) Given $f_{bw}(x_i) = \frac{g_{bw}(x_i)}{T_{bw}}$ it is easy to see that if $g_{bw}(x_i) = K_{bw}\ \forall x_i \in X_{bw}$ then $T_{bw} = \sum_{i=1}^{n} g_{bw}(x_i) = n \cdot K_{bw}$ and, as a consequence, $f_{bw}(x_i) = \frac{K_{bw}}{n \cdot K_{bw}} = \frac{1}{n}\ \forall x_i \in X_{bw}$. Hence, by replacing $f_{bw}(x_i) = \frac{1}{n}$ in Equation (5.2), we get $d_{bw} = 1$. □

Figure 5.2 shows the relation between the uniformity of the bandwidth of the nodes and the anonymity degree of the whole system. It depicts the anonymity degree of a Tor system with 100 nodes, measured under different restrictions. In particular, the bandwidth of the nodes has been modified in a manner that a certain subset of nodes has the same bandwidth, and the bandwidth of the remainder nodes has been fixed at random. During all the measurements the total bandwidth of the system T_{bw} remains constant. As the size of the subset is increased, and more nodes have the same bandwidth, the uncertainty is higher from the point of view of the discrete random variable associated to $\psi_{bw}(N, \delta)$. Therefore, the anonymity degree is increased when the uniformity of the distribution of the bandwidths grows.

5.3 Conclusion

In this chapter we have presented a formal model of the Tor node selection criteria. It provides a definition of the selection of Tor nodes process, of the corresponding adversary model proposed by Syverson *et al.* targeting the communication anonymity of Tor users, and an analytical expression to compute the anonymity degree of the Tor infrastructure based on the circuit construction criteria. In order to compute the degree of anonymity, the formal model maps a selection algorithm to a discrete random variable with its corresponding probability mass function. Then, the normalised Shannon entropy is applied to this discrete random variable. Also, we have illustrated how this formal model can be applied to some particular node selection criterion (*i.e.* random selection, geographical selection, and bandwidth selection), and how it can be useful to compare different algorithms and to infer other underlying properties.

6

Latency graphs for the Tor circuit construction

> *A careful analysis of the process of observation in atomic physics has shown that the subatomic particles have no meaning as isolated entities, but can only be understood as interconnections between the preparation of an experiment and the subsequent measurement.*
>
> ERWIN SCHRRÖDINGER

In Chapter 4 we have seen how the Tor network can be a useful tool as a way to preserve the privacy of the DNS protocol. In spite of this, in Chapter 5 we have argued that there is a tight link between the degree of anonymity and the strategy used for the selection of nodes. Also, we have suggested that the selection criteria influence in the network latencies when a user anonymise its traffic through Tor. This conducts us to a trade-off between degree of anonymity and network latencies caused by the way the nodes of a circuit are chosen. We present in this chapter a new selection algorithm based on the concept of *latency graphs*. This algorithm aims at reducing the success probability of linking attacks while providing enough performance for low-latency services. A series of experiments, conducted on a real-world Tor deployment over PlanetLab, confirm the validity of the new strategy, and show its superiority over other classical ones.

6.1 New Tor selection strategy based on latency graphs

The new strategy relies on modelling the Tor network as an undirected graph $G(V, E)$, where $V = N \cup \{s\}$ denotes the set composed by the Tor nodes $N = \{v_1, ..., v_n\}$ and the client node $v_{n+1} = s$, and where $E = \{e_{12}, e_{13}, ..., e_{ij}\}$ denotes the set of the edges of the graph. We use the notation $e_{ij} = (v_i, v_j)$ to refer to the edge between two nodes v_i and v_j. The set of edges E represents the potential connectivity between the nodes in V, according to some partial knowledge of the network status which the strategy has. If an edge $e_{ij} = (v_i, v_j)$ is in E, then the connectivity between nodes v_i and v_j is potentially possible. The set of edges E is a dynamic set, *i.e.*, the network connectivity (from a TCP/IP standpoint) changes periodically in time, while the set of vertices V is a static set. Finally, and although the network connectivity from node v_i to v_j is not necessarily the same as the connectivity from v_j to v_i, we decided to model the graph as undirected for simplicity reasons. Our decision also obeys to the two following facts: (i) in a TCP/IP network, the presence of nodes is more persistent than the connectivity among them; and (ii) the connectivity is usually the same from a bidirectional routing point of view in TCP/IP networks.

Related to the edges of the graph $G(V, E)$, we define a function $c_t : E \to \mathbb{R} \cup \{\infty\}$ such that, for every edge $e_{ij} \in E$, the function returns the associated network latency between nodes v_i and v_j at time t. If there is no connectivity between nodes v_i and v_j at time t, then we say that the connectivity is undefined, and function c_t returns the infinity value. Notice that function c_t can be implemented in several ways and, according to Coates *et al.* [46], there is some previous work in the field of network measurement that could be used. This previous research includes software tools to monitor/probe the network, probabilistic modelling of network queues, inference from measurements of streams of traffic, or network tomography. Regardless of the strategy used to implement c_t, there is an important restriction from a security point of view: leakage of sensitive information in the measurement process shall be contained. This mandatory constraint must always be fulfilled. Otherwise, an adversary can benefit from a monitoring process in order to degrade the anonymity degree.

Given the aforementioned rationale, we propose now the construction of our new selection strategy by means of two general processes. A first process computes and maintains the set of edges of the graph and its latencies. The second process establishes, according to the outcomes provided by the first process, circuit nodes. Circuit nodes are chosen from those identified within graph paths with minimum latency. These two processes are summarised, respectively, in Algorithms 6.1 and 6.3. A more detailed explanation of the proposed strategy is given below.

The first process (*cf.* Algorithm 6.1) is executed in background and keeps a set of labels related to each edge. Every label is defined by the expression $L(e_{ij}) = (l, t)$, where e_{ij} denotes its associated edge. The label contains a tuple (l, t) composed by an estimated latency l between the nodes of the edge (*i.e.*, v_i and v_j), and a time instant t which specifies when the latency l was computed. When the process is executed for the first time, the set of edges and all the labels are initialised as $E \leftarrow \varnothing$ and $L(e_{ij}) \leftarrow (\infty, 0)$.

At every fixed interval of time Δt, the process associated to Algorithm 6.1 proceeds indefinitely as follows. A set of m edges associated to the complete graph K_n with the same vertices of $G(V, E)$ are chosen at random. The latency associated to every edge is estimated by means of the aforementioned

Algorithm 6.1 Latency Computation Process - lat_comp$(G(V,E), \Delta t, m)$

Input: $G(V,E), \Delta t, m$

$t_0 \leftarrow t_q \leftarrow 0$
$E \leftarrow \varnothing$
$L(e_{ij}) \leftarrow (\infty, t_0)$

while *TRUE* **do**
  $t_q \leftarrow t_q + 1$
  for $i \leftarrow 1$ **to** m **do**
    $i, j \leftarrow$random$(1, |V|)$, $i \neq j$
    $l_q \leftarrow c_t(e_{ij})$
    if $l_q = \infty$ **then**
      $E \leftarrow E \setminus \{e_{ij}\}$
    else
      $E \leftarrow E \cup \{e_{ij}\}$
      Given $L(e_{ij}) = (l_p, t_p)$
      if $l_p \neq \infty$ **then**
        $\alpha \leftarrow (t_p - t_0)/(t_q - t_0)$
        $l_q \leftarrow \alpha \cdot l_p + (1 - \alpha) \cdot l_q$
      end if
      $L(e_{ij}) \leftarrow (l_q, t_q)$
    end if
  end for
  sleep(Δt)
end while

function c_t. If the computed latency is undefined (*i.e.*, function c_t returns the infinity value), then the edge is removed from the set E (if it was already in E) and the associated latency labels not updated. Otherwise, the edge is added to the set E (if it was not already in E), and the value of its corresponding labels updated. In particular, the latency member of the tuple is modified by using an Exponentially Weighted Moving Average (EWMA) strategy [76], and the time member is updated according to the current time instant t_q. For instance, let us suppose that we are in the time instant t_q and we have chosen randomly the edge e_{ij} with an associated label $L(e_{ij}) = (l_p, t_p)$. Let us also suppose that $l_q = c_{t_q}(e_{ij})$ is the new latency estimated for such an edge. Thus, its corresponding label is updated according to the following expression:

$$L(e_{ij}) \leftarrow \begin{cases} (l_p, t_p), & \text{if } l_q = \infty; \\ (l_q, t_q) & \text{if } l_p = \infty; \\ (\alpha \cdot l_p + (1-\alpha) \cdot l_q, t_q) & \text{otherwise} \end{cases}$$

The first case of the previous expression corresponds to a situation of disconnection between the nodes of the edge e_{ij}, and that has been detected by the function c_{t_q}. As a consequence, $c_{t_q}(e_{ij})$ returns infinity. In this case, the previous estimated latency l_p is maintained in the tuple, and the edge e_{ij} is removed from E. The second case can be associated to the first time the latency of the edge e_{ij} is estimated using c_{t_q}, since the previous latency was undefined and the infinity value is the one used in the first

Algorithm 6.2 K-paths Computation Process - kpaths($G(V,E), \delta, k, x\_node, cur\_path, paths\_list$)

Input: $G(V,E), \delta, k, x\_node, cur\_path, paths\_list$

if len($paths\_list$) $= k$ **then**
    return
end if
if len(cur_path) $> \delta$ **then**
    return
end if

$v_l \leftarrow$last_vertex(cur_path)
$new_len \leftarrow$len(cur_path)+1
$adjacency_list \leftarrow$adjacent_vertices($G(V,E)$, v_l)
remove_nodes($adjacency\_list$, cur_path)
random_shuffle($adjacency\_list$)

for $vertex$ **in** $adjacency\_list$ **do**
    if $vertex = x\_node$ **and** $new\_len < \delta$ **then**
        continue
    end if
    if $vertex = x\_node$ **and** $new\_len = \delta$ **then**
        $new\_sol \leftarrow cur\_path + \langle vertex \rangle$
        $paths\_list \leftarrow paths\_list + \langle new\_sol \rangle$
        break
    end if
    $cur\_path \leftarrow cur\_path + \langle vertex \rangle$
    kpaths($G(V,E), \delta, k, x\_node, cur\_path, paths\_list$)
end for

instantiation of $L(e_{ij})$. Under the two last cases of the previous expression, the edge e_{ij} is always added to the set E if it still does not belong to the aforementioned set. The third scenario corresponds to the EWMA in the strict sense. In this case, the coefficient $\alpha \in (0,1)$ represents a smoothing factor. The value α has an important effect in the resulting estimated latency stored in $L(e_{ij})$. Notice that those values of α that are close to zero give a greater weight to the recent measurements of the latency through the function c_{t_q}. Contrary to this, a value of α closer to one gives a greater weight to the historical measurements, making the resulting latency less responsive to recent changes.

For the definition of the α factor we must consider that the previous update of the latency —for a certain edge— could have been performed long time ago. This is possible since, for every interval of time Δt we choose randomly just only m edges to update their latencies. Indeed, the value of l_p in the previous example could have been computed at the time instant t_p, and where $t_p \ll t_q$. Therefore, if we define α as a static value, the weight for previous measurements will always be the same, independently of when the measurement was taken. This is not an acceptable approach since the older the previous measurement is, the less weight should have in the resulting computed latency.

To overcome this semantic problem, the coefficient α must be defined as a dynamic value that takes into account the precise moment in which the previous latencies were estimated for every edge. In other words, α should be inversely proportional to the size of the time interval between the previous

Algorithm 6.3 Graph of Latencies Selection of Nodes - $\psi_{grp}(N, \delta)$

Input: $G(V, E), s, \delta, k, max\_iter, \Delta t$
Output: $C = \langle s, e, r_1, r_2, ..., r_{\delta-2}, x \rangle$, $P = \{a1, ..., a_\delta\}$

$P \leftarrow \varnothing$
$paths\_list \leftarrow \langle\rangle$
$iter \leftarrow 0$

/ Executed in background as a process */*
lat_comp$(G(V, E), \Delta t, m)$

repeat
    $cur\_path \leftarrow \langle s \rangle$
    $x_node \leftarrow$random_vertex$(V \setminus \{s\})$
    kpaths$(G(V, E), \delta, k, x_node, cur_path, paths_list)$
    $iter \leftarrow iter + 1$
until (**not** empty$(paths_list)$) **or** $(iter = max\_iter)$

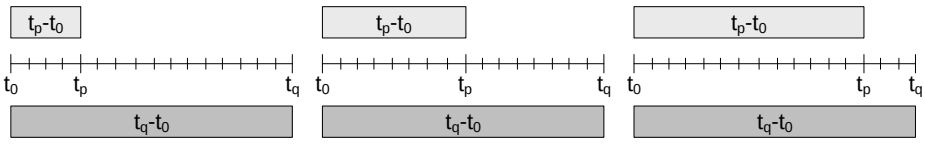
if not empty$(paths_list)$ **then**
    $C \leftarrow$min_weighted_path$(paths_list)$
else
    $C \leftarrow$random_path(V, δ)
end if
for $i \leftarrow 1$ **to** $\delta - 1$ **do**
    $P \leftarrow P \cup \{(c_i, c_{i+1})\}$
end for

measurement and the current one. In order to define α as a function of this time interval, we must keep the time instant of the previous latency estimation for a given edge. This can be accomplished by storing the time instants in the tuple of every edge label. Hence, every time we select at random m edges to update their latencies, its associated time members of its labels must be updated with the current time instant t_q. It is important to remark that this update process must be done just only when the function c_t returns a value different from the infinity one. Moreover, for a selected edge e_{ij} in the time instant t_q, its α value is defined as:

$$\alpha = \frac{t_p - t_0}{t_q - t_0}$$

where t_0 is the first time instant when the execution of the process started. A graphical interpretation of the previous expression is depicted in Figure 6.1. We can appreciate that $\alpha \in (0, 1)$ by associating the numerator and the denominator of the expression with its interval representation in the figure. Thus, we can directly deduce that $0 < (t_p - t_0) < (t_q - t_0)$ and, consequently, $\alpha \in (0, 1)$. In this figure, we can also see the influence of the previous time instant t_p on the resulting α. In particular, three cases are presented: a) $t_p \ll t_q$, b) $t_p \approx \frac{t_q - t_0}{2}$, and c) $t_p \approx t_q$. For these cases, we can observe how α tends to, respectively, 0, 0.5 and 1.

The second process (*cf.* Algorithm 6.3) is used for selection of circuit nodes. It utilises the information maintained by the process associated to Algorithm 6.1. In particular, the graph $G(V, E)$ and the labels $L(e_{ij})\ \forall e_{ij} \in E$ are shared between both processes. When a user wants to construct a new circuit, this process is executed and it returns the nodes of the circuit. For this purpose, an exit node x is chosen at



(a) $\alpha \to 0$ **(b)** $\alpha \to 0.5$ **(c)** $\alpha \to 1$

Figure 6.1: Graphical interpretation of the α coefficient

random from the set of vertices $V \setminus \{s\}$. After that, the process computes until k random paths of length δ between the nodes s and x. With this aim, a recursive process, summarised in Algorithm 6.2, is called. In the case that there is not any path between the vertices s and x, another exit node is chosen and the procedure is executed again. This iteration must be repeated until a) some paths of length δ between the pair of nodes s and x are found, or b) until a certain number of iterations are performed. In the first case, the path with the minimum latency is selected as the solution among all the obtained paths. In the second case, a completely random path of length δ is returned. To avoid this situation, *i.e.*, to avoid that our new strategy behaves as a random selection of nodes strategy, the process associated to Algorithm 6.1 must be started some time before the effective establishment of circuits take place. This way, the graph $G(V, E)$ increases the necessary level of connectivity among its vertices. We refer to Section 6.3 for more practical details and discussions on this point.

6.1.1 Discussion on the adversary model

One may think that an adversary, as it was initially defined in Chapter 5, Section 5.1.2, can try to reconstruct the client graph and guess the corresponding latency labels of our new strategy in order to degrade its anonymity degree. However, even if we assume the most extreme case, in which the adversary obtains a complementary complete graph K_n with the set of vertices N and corresponding latency labels, this does not affect the anonymity degree of our new strategy. First of all, we recall that the graph of the client is a dynamic random subgraph of K_{n+1} that is evolving over time, with a set of vertices $N \cup \{s\}$. The adversary graph would also be a subgraph of K_n with the set of vertices N, changing dynamically as time goes by. Therefore, the set of vertices and edges of the adversary and client graphs will never converge into same connectivity model of the network. Moreover, the latencies between the client node s and any other potential entry node e cannot be calculated by the adversary. Otherwise, this would mean that the anonymity has already been violated by the adversary. Indeed, the estimated latencies will definitively differ between the client and the adversary graph, since they are computed at different time frames and different source networks. Finally, the adversary also ignores the exit nodes selected by the client, as well as the k parameter used by the client to choose the paths.

6.2 Analytical evaluation of the new strategy

We provide in this section the analytical expression of the anonymity degree of the new strategy. First, we extend the list of definitions provided in Chapter 5.

6.2.1 Analytical graph of $\psi_{grp}(N, \delta)$

In order to provide an analytical expression of the anonymity degree it is important to notice that this must always be done from the adversary standpoint. In this regard, the graph to be considered for this purpose differs with respect to the one used to compute a circuit. Note that the latencies associated to every edge which contains the client node s cannot be estimated by the adversary — specially if we consider that this particular node is unknown by the adversary. Hence, an adversary who wants to compromise the anonymity of the client node s could try to estimate the user graph without the node s and its associated edges. This leads us to the following definition (*cf.* Figure 6.2 as a clarifying example):

Definition 27. *Given a latency graph $G(V, E)$ associated to a selection of Tor nodes $\psi_{grp}(N, \delta)$ strategy and the client node s, we define the analytical graph as $G'(V', E')$ where $V' = V \setminus \{s\}$ and $E' = E \setminus \{(s, v_i)\}$ $\forall v_i \in V$.*

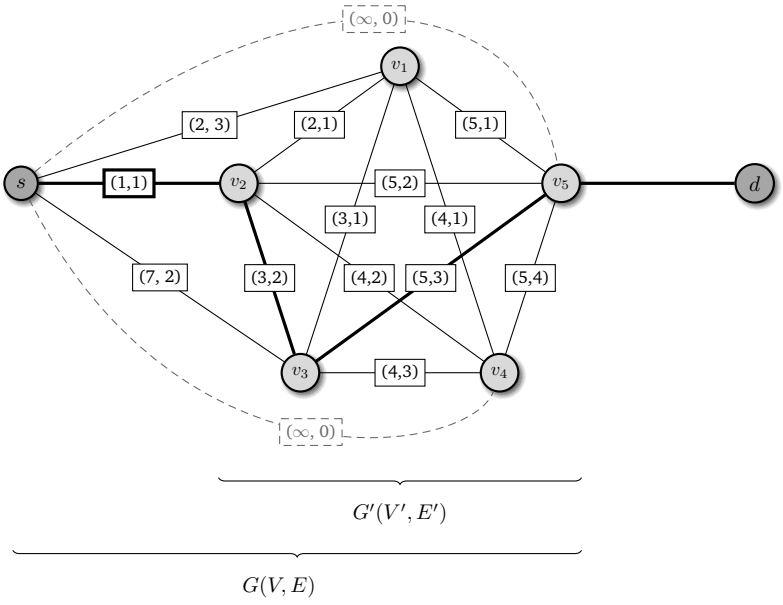


Figure 6.2: Example of a latency graph and its analytical graph with a selected circuit $C := \langle s, v_2, v_3, v_5 \rangle$ of length $\delta := 3$

6.2.2 λ-betweenness and λ-betweenness probability

For the purpose of computing the degree of anonymity of our new strategy, a new metric inspired by the Freeman's *betweenness centrality* measure [61] is presented. This metric, called λ-betweenness, is defined as a measurement of the frequency which a node v is traversed by all the possible paths of length λ in a graph. The formal definition is given below.

Definition 28. *Consider an undirected graph $G(V, E)$. Let KP_{st} denote the set of paths of length λ between a fixed source vertex $s \in V$ and a fixed target vertex $t \in V$. Let $KP_{st}(v)$ be the subset of KP_{st} consisting of*

paths that pass through the vertex v. Then, we define the λ-betweenness of the node $v \in V$ as follows:

$$KP_B(v,\lambda) = \frac{\sum\limits_{s,t \in V} \sigma_{st}(v,\lambda)}{\sum\limits_{s,t \in V} \sigma_{st}(\lambda)}$$

where $\sigma_{st}(\lambda) = |KP_{st}|$ and, $\sigma_{st}(v,\lambda) = |KP_{st}(v)|$.

As we can observe, the λ-betweenness provides the proportion between the number of paths of length λ which traverses a certain node v, and the number of the total paths of length λ. However, since the degree of anonymity needs a probability distribution, the following definition is required.

Definition 29. *Consider an undirected graph $G(V,E)$. Let $KP_B(v,\lambda)$ be the λ-betweenness of the node $v \in V$. Then, the λ-betweenness probability of the node v is defined as:*

$$LB(v,\lambda) = \frac{KP_B(v,\lambda)}{\sum\limits_{w \in V} KP_B(w,\lambda)} = \frac{\sum\limits_{s,t \in V} \sigma_{st}(v,\lambda)}{\sum\limits_{w \in V} \sum\limits_{s,t \in V} \sigma_{st}(w,\lambda)}$$

It follows immediately that $0 \leq LB(v,\lambda) \leq 1$, $\forall v \in V$, since this expression is equivalent to the normalised λ-betweenness.

6.2.3 Entropy and anonymity degree

Given the previous definitions, and by using the formal model presented in Chapter 5, we can now obtain the analytical expression for the degree of anonymity of this new strategy. Thus, the graph of latencies selection of Tor nodes is defined formally as an algorithm $\psi_{grp}(N,\delta) \sim f_{grp}(x)$ with an associated discrete random variable X_{grp} and an analytical graph $G'(V',E')$. The pmf $f_{grp}(x)$ is given by means of the λ-betweenness probability expression:

$$f_{grp}(x_i) = p_i = \mathbb{P}(X_{grp} = x_i) = \frac{\sum\limits_{e,x \in V'} \sigma_{ex}(v_i,\lambda)}{\sum\limits_{w \in V'} \sum\limits_{e,x \in V'} \sigma_{ex}(w,\lambda)}$$

where e and x denotes every potential entry and exit node respectively in a Tor circuit, and $\lambda = \delta - 1$. It is worth noting that the value $\lambda = \delta - 1$ makes sense only if we take into consideration that the client node s and its edges are removed in the analytical graph respect to the latency graph.

Hence, the entropy of a system whose clients use a graph of latencies selection of nodes strategy is:

$$H_{grp}(X) = -\sum_{i=1}^{n} LB(v_i,\lambda) \cdot log_2\big(LB(v_i,\lambda)\big)$$

By replacing $H_{grp}(X)$ in Equation (5.2), the degree of anonymity is then:

$$d_{grp} = -\sum_{i=1}^{n} \frac{LB(v_i,\lambda)}{log_2(n)} \cdot log_2\big(LB(v_i,\lambda)\big)$$

Theorem 8. *Given a selection of Tor nodes $\psi_{grp}(N,\delta) \sim f_{grp}(x)$ with an associated discrete random variable X_{grp} and an analytical graph $G'(V',E')$ with $n = |V'|$ and $m = |E'|$, the anonymity degree is increased as the density of the analytical graph grows.*

Proof. The density of an analytical graph $G' = (V', E')$ measures how many edges are in the set E' compared to the maximum possible number of edges between vertices in the set V'. Formally speaking, the density is given by the formula $\frac{2m}{n(n-1)}$. According to the previous expression, and since the number of nodes of the analytical graph remains constant, the only way to increase the density value is through rising the value m; that is, by adding new edges to the graph. Obviously, this implies that the more number of edges the analytical graph has, the more its density value is augmented.

Moreover, if we increase the density of the analytical graph by adding new edges, then the λ-betweenness probability of each vertex will be affected. In particular, the denominator of the λ-betweenness probability expression will change for all the vertices in the same manner, whereas the numerator will be increased for those vertices that lie on any new path of length λ which contains some of the added edges. However, this increase is not arbitrary for a given vertex, since it has a maximum value determined by the total amount of paths of length λ which traverses such vertex. Therefore, we can consider that each vertex has two states while we are adding new edges. First, a transitory state where the graph does not include all the paths of length λ that traverse such vertex. And second, a stationary state which implies that the graph has all the paths of length λ that traverses the given vertex. Thus, if we add new edges at random, then the numerator of the λ-betweenness probability of each vertex should be increased uniformly. Consequently, the degree of anonymity grows when the density of the graph is augmented. □

It is interesting to highlight that the numerator of the λ-betweenness probability of a certain vertex will be increased while it is in a transitory state, and until the vertex achieves its stationary state. After that, such value cannot be increased. It seems obvious that the degree of anonymity associated to a particular analytical graph will be reached when all the vertices are in a stationary states; or, in other words, when it is the complete graph. Let us formalise this through the following theorem.

Theorem 9. *Given a selection of Tor nodes $\psi_{grp}(N,\delta) \sim f_{grp}(x)$ with an associated discrete random variable X_{grp} and an analytical graph $G'(V',E')$ with $n = |V'|$, the maximum anonymity degree is achieved iff $G'(V',E')$ is the complete graph K_n.*

Proof. ($\Rightarrow$) Let us suppose that $G'(V',E')$ is not the complete graph K_n. The maximum anonymity degree will be achieved when $LB(v_i,\lambda)$ is equiprobable for all $v_i \in V'$. That is:

$$\frac{\sum\limits_{e,x\in V} \sigma_{ex}(v_i,\lambda)}{\sum\limits_{w\in V}\sum\limits_{e,x\in V} \sigma_{ex}(w,\lambda)} = \frac{1}{n} \quad \forall v_i \in V'$$

where $\lambda = \delta - 1$, and where e and x represents every possible *entry* and *exit* node of a circuit respectively. The previous expression can be rewritten as follows:

$$\sum_{e,x\in V'} \sigma_{ex}(v_i,\lambda) = \frac{\sum\limits_{e,x\in V'} \sigma_{ex}(v_1,\lambda) + ... + \sum\limits_{e,x\in V'} \sigma_{ex}(v_n,\lambda)}{n}$$

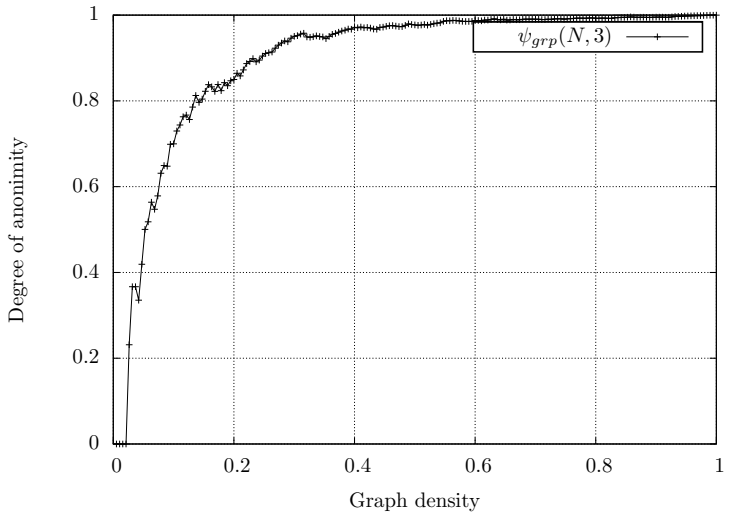


Figure 6.3: Influence of the density of the analytical graph in the degree of anonymity with $|V'| = 20$ and $\delta = 3$

Let us now suppose that the value $\sum_{e,x \in V'} \sigma_{ex}(v_i, \lambda)$ is fixed for every node of the analytical graph in accordance to the previous expression. Then, since $G'(V', E')$ is not the complete graph K_n, we can eliminate an arbitrary edge such that the number of paths of length λ with entry node e and exit node x, and which traverses a given particular node $v_j \in V'$, is reduced. Thus, the value of $\sum_{e,x \in V'} \sigma_{ex}(v_j, \lambda)$ would be affected for that given node. However, this contradicts the previous expression, since $\sum_{e,x \in V'} \sigma_{ex}(v_i, \lambda)$ would take different values for distinct nodes, and when such value must be the same for any node of the graph.

($\Leftarrow$) Let us suppose, by contradiction, that the maximum anonymity degree is not achieved by the analytical graph K_n associated to $\psi_{grp}(N, \delta)$. This implies that given two different nodes v_j and v_k of the graph K_n, they will not have the same probability of being chosen by $\psi_{grp}(N, \delta)$; that is, $LB(v_j, \lambda) \neq LB(v_k, \lambda)$. Then, since $LB(v, \lambda)$ is defined as follows:

$$LB(v, \lambda) = \frac{\sum\limits_{e,x \in V'} \sigma_{ex}(v, \lambda)}{\sum\limits_{w \in V'} \sum\limits_{e,x \in V'} \sigma_{ex}(w, \lambda)}$$

we can consider that the only factor which makes possible the previous restriction $LB(v_j, \lambda) \neq LB(v_k, \lambda)$ is in the numerator, because the value of the denominator remains equal for both nodes in a fixed graph. Thus, if we want to satisfy the previous restriction, we must change the value $\sum_{e,x \in V'} \sigma_{ex}(v, \lambda)$ of either node v_j or node v_k. However, this is only possible if we eliminate a particular edge of the graph. This contradicts the imposed premise that the analytical graph associated to $\psi_{grp}(N, \delta)$ was the complete graph K_n. □

Theorems 8 and 9 are exemplified in conjunction in Figure 6.3. We can observe how a density increase of an analytical graph influences in the degree of anonymity, achieving its maximum value when the graph is the complete one (*i.e.*, it has a density equal to one).

Theorem 10. *Let $G(V, E)$ be a undirected graph with $n = |V|$ and let λ be a fixed length of a path, the value of $\sigma_{st}(\lambda)$ is maximised iff $G(V, E)$ is the complete graph K_n.*

Proof. ($\Rightarrow$) Let us suppose, by contradiction, that $G(V, E)$ is not the complete graph K_n. Then, we can choose an arbitrary edge $e_{ij} \in E$ that belongs to a path of length λ between the nodes s and t. Then, we can remove e_{ij} from E since the graph is not complete. As a consequence, the value KP_{st} will be reduced. However, this contradicts the fact that the value $\sigma_{st}(\lambda)$ must be maximum since $\sigma_{st}(\lambda) = |KP_{st}|$.

($\Leftarrow$) The proof is direct, since the complete graph K_n contains all the possible edges between its nodes, and thus KP_{st} consists of all the possible paths of length λ between the nodes s and t. □

Theorem 11. *Let K_n be a complete graph, the total number of paths of length λ between any pair of vertices s and t is given by the expression:*

$$\sum_{s,t \in V} \sigma_{st}(\lambda) = ((n-1)((n-1)^{\lambda} - (-1)^{\lambda}))$$

Proof. The proof is given in Appendix A. □

Theorem 12. *Given a selection of Tor nodes $\psi_{grp}(N, \delta) \sim f_{grp}(x)$ with an associated discrete random variable X_{grp} and an analytical graph $G'(V', E')$, the maximum anonymity degree is achieved iff*

$$\sum_{e,x \in V'} \sigma_{ex}(\lambda) = ((n-1)((n-1)^{\lambda} - (-1)^{\lambda}))$$

Proof. The proof is direct by applying Theorems 9, 10 and 11. □

6.3 Experimental results

We present in this section a practical implementation and evaluation of the series of strategies previously exposed (*cf.* Chapter 5, Section 5.2). Each implementation has undergone several tests, in order to evaluate latency penalties during Web transmissions. Additionally, the degree of anonymity of every experimental test is also estimated, for the purpose of drawing a comparison among them.

6.3.1 Node distribution and configuration in PlanetLab

In order to measure the performance of the strategies presented in our work, some practical experiments have been conducted. In particular, we deployed a private network of Tor nodes over the PlanetLab research network [43, 113]. Our deployed Tor network is composed of 100 nodes following a representative distribution based on the real (public) Tor network. We distributed the nodes of the private Tor network following the public network distribution in terms of countries and bandwidths. Table 6.1 summarises the distribution values per country. The estimated bandwidths of the nodes is retrieved through the Directory Servers of the real Tor network [53]. Then, we categorised the nodes according to their bandwidths by means of the *k-means clustering* methodology [7, 92]. A value of $k = 100$ is used as the number of clusters (*i.e.*, number of selected nodes in PlanetLab). When the algorithm converges,

Real Tor network			PlanetLab
# Nodes	Country	%	# Nodes
815	US	26.54	27
533	DE	17.36	17
187	RU	6.09	6
181	FR	5.89	6
171	NL	5.56	6
146	GB	4.75	5
132	SE	4.30	4
80	CA	2.61	3
56	AT	1.82	2
43	AU	1.40	1
40	IT	1.30	1
40	UA	1.30	1
39	CZ	1.27	1
38	CH	1.24	1
34	FI	1.11	1
34	LU	1.11	1
33	PL	1.08	1
32	JP	1.04	1
437	Others (<1%)	14.23	15
3071	–	100	100

Table 6.1: Selected PlanetLab nodes per country according to the real Tor network distribution

a cluster is assigned randomly to each node of the private Tor network. Subsequently, the bandwidth of each node is configured with the value of its associated centroid (*i.e.* the mean of the cluster). For such a purpose, the directive `BandwidthRate` is used in the configuration file of every node. Let us note that the country and bandwidth values are considered as independent in the final node distribution configuration. Indeed, there is no need to correlate both variables, since the bandwidth of every node can be configured by its corresponding administrator, while this fact does not depend on the country which the node belongs to.

6.3.2 Testbed environment

Every node of our Planetlab private network runs the Tor software, version 0.2.3.11-alpha-dev. Additionally, four nodes inside the network are configured as Directory Servers. These four nodes are in charge of managing the global operation of the Tor network and providing the information related to the network nodes.

Furthermore, two additional nodes outside the PlanetLab network are used in our experiments. One of them is based on an Intel Core2 Quad Processor at 2.66GHz with 6GB of RAM and a Gentoo GNU/Linux Operating System with a 3.2.9 kernel. This one is used as the *client* node who handles the construction of Tor circuits for every evaluated strategy. For this purpose, this node also runs our own specific software application, hereinafter denoted as `torspd.py`. A beta release of `torspd.py`, written in

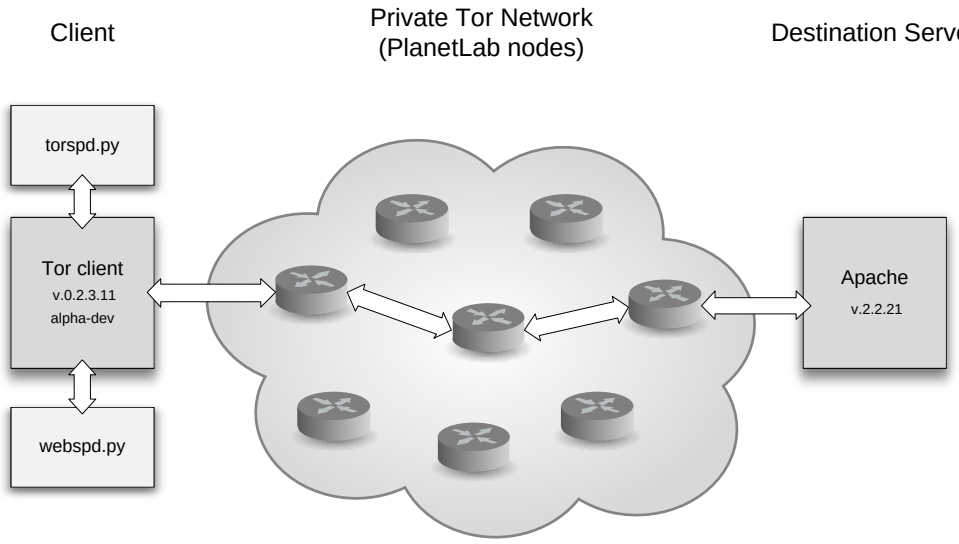
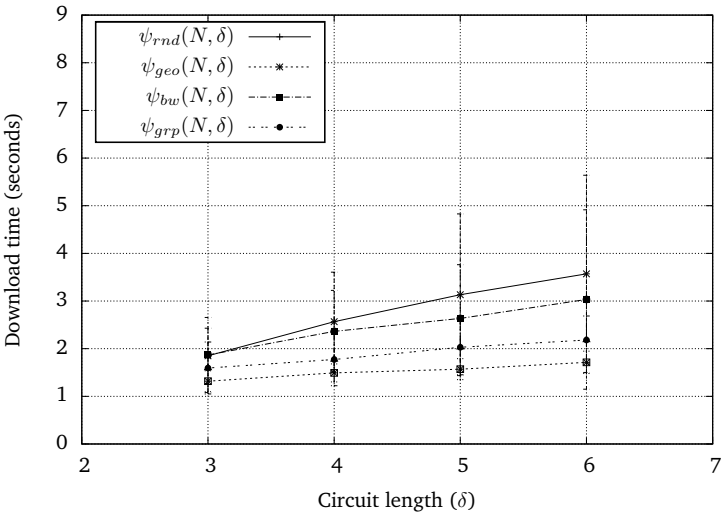
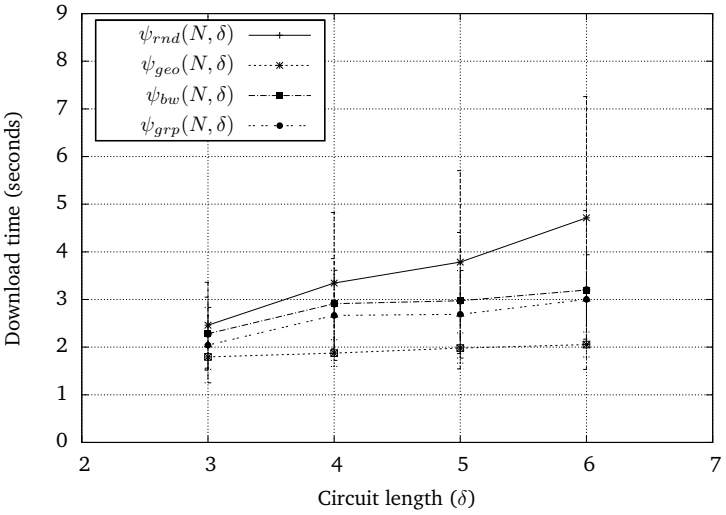


Figure 6.4: Conceptual representation of our testbed environment

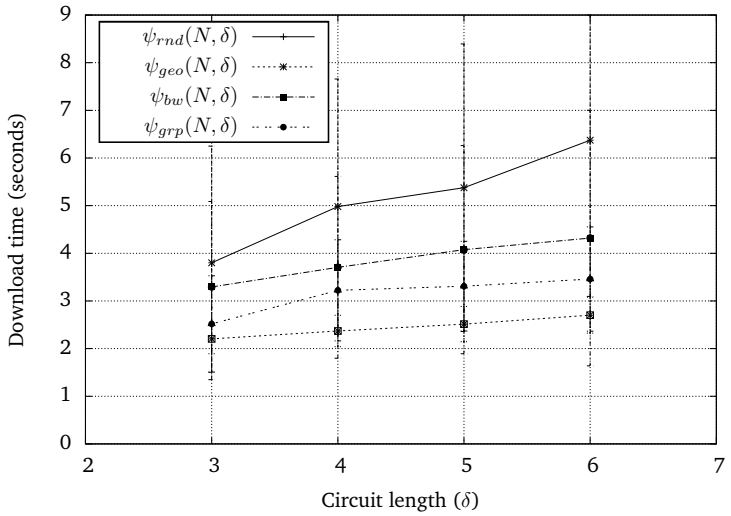
Python 2.6.6, can be downloaded at `https://github.com/sercas/torspd`. The `torspd.py` application relies on the TorCtl Python bindings [94] —a Tor controller software to support path building and various constraints on node and path selection, as well as statistic gathering. Moreover, `torspd.py` also benefits from the package NetworkX [70] for the creation, manipulation, and analysis of graphs. The client node is not only in charge of the circuit construction given a certain strategy, but also of attaching an initiated HTTP connection to an existing circuit. To accomplish this, the node uses `torspd.py` to connect to a special port of the local Tor software called the *control port*, and which allows to command the operations. The client node includes an additional software —also based on Python— capable of performing HTTP queries through our private Tor network by using a SOCKS5 connection against the local Tor client. This software, called `webspd.py`, is also able to obtain statistics results about the launched queries in order to evaluate the performance of the algorithms implemented in `torspd.py`. Finally, `webspd.py` performs every HTTP query making use directly of the IP address of the destination server; consequently, any perturbation introduced by a DNS resolution is avoided in our measurements. The second node outside the PlanetLab network is based on an Intel Xeon Processor at 2.00GHz with 2GB of RAM and a Debian GNU/Linux Operating System with a 2.6.26 kernel. This node is considered as the *destination server*, and includes an HTTP server based on Apache, version 2.2.21. The conceptual infrastructure used to carry out our experiments is illustrated in Figure 6.4.



(a) Web size of 50KB



(b) Web size of 150KB



(c) Web size of 320KB

Figure 6.5: Experimental results

$\psi_{rnd}(N,\delta)$, $d_{rnd} = 1.0$, websize 50KB				
Circ. length	Min.	Max.	Avg.	Std. dev.
$\delta = 3$	0.95094203949	3.38077807426	1.84956678152	0.58107725003
$\delta = 4$	1.14792490005	7.46992301941	2.56735023022	1.03927644851
$\delta = 5$	1.13161778450	12.7252390385	3.13187572718	1.69722167190
$\delta = 6$	1.57145905495	14.6901309490	3.56973065615	2.06960596616
$\psi_{rnd}(N,\delta)$, $d_{rnd} = 1.0$, websize 150KB				
Circ. length	Min.	Max.	Avg.	Std. dev.
$\delta = 3$	0.970992088318	5.70451307297	2.46016269684	0.901269612931
$\delta = 4$	1.081045866010	12.0326070786	3.34545367479	1.478886535440
$\delta = 5$	1.624027013780	16.0551090240	3.78437126398	1.918732505410
$\delta = 6$	2.279263019560	11.5805990696	4.71352141102	2.544477101520
$\psi_{rnd}(N,\delta)$, $d_{rnd} = 1.0$, websize 320KB				
Circ. length	Min.	Max.	Avg.	Std. dev.
$\delta = 3$	1.49153804779	13.2033219337	3.79921305656	2.45165379541
$\delta = 4$	1.84271001816	15.2616338730	4.98011079788	2.67792560196
$\delta = 5$	1.73619008064	17.1969499588	5.37626729012	3.01781647919
$\delta = 6$	2.16737580299	17.8402540684	6.37420113325	3.27889183837

Table 6.2: Experimental results, table ψ_{rnd}

With the purpose of obtaining extrapolative results, we consider in our testbed the outcomes reported in [114]. This report, based on the analysis of more than four billion Web pages, provides estimations of the average size of current Internet sites, as well as the average number of resources per page and other interesting metrics. Our testbed is built bearing in mind these premises, so that it is close enough to a real Web environment. This way, the analysed strategies (*i.e.*, random selection, geographical selection, bandwidth selection, and graph of latencies selection) are evaluated based on three different series of experiments that vary the Web page sizes. More precisely, the client node requests via our private PlanetLab Tor network Web pages of, respectively, 50KB, 150KB and 320KB of size —being the last one the average size of a Web page according to the aforementioned report. The length of the circuits is seen as another variable in our testbed. More precisely, the different strategies are evaluated with Tor circuits of length three, four, five and six. Every experiment is repeated 100 times, from which we obtain the minimum, maximum and average time needed to download the corresponding Web pages. Likewise, the standard deviation is computed for every test. The obtained numerical results are presented in Tables 6.2, 6.3, 6.4 and 6.5, and also depicted graphically in Figure 6.5. In the sequel, we use these results to analyse the performance of every strategy in terms of transmission times and degree of anonymity.

6.3.3 Random selection of nodes strategy evaluation

As previously exposed in Theorem 3 (*cf.* Chapter 5, Section 5.2.1), the random selection of nodes strategy is the best one from the point of view of the degree of anonymity, since it achieves the maximum possible

$\psi_{geo}(N,\delta)$, $d_{geo} \approx 0.7157$, websize 50KB				
Circ. length	Min.	Max.	Avg.	Std. dev.
$\delta = 3$	0.913872003555	2.36748099327	1.31694087505	0.219359721740
$\delta = 4$	1.083739995960	2.03739213943	1.49165359974	0.189194613865
$\delta = 5$	1.157481908800	2.17184281349	1.56993633509	0.220167861127
$\delta = 6$	1.200492858890	2.63958501816	1.71368015051	0.234977785757
$\psi_{geo}(N,\delta)$, $d_{geo} \approx 0.7157$, websize 150KB				
Circ. length	Min.	Max.	Avg.	Std. dev.
$\delta = 3$	1.38168692589	2.68786311150	1.79467165947	0.260276001481
$\delta = 4$	1.27939105034	2.92536497116	1.87463890314	0.281488220772
$\delta = 5$	1.33843898773	3.71059083939	1.98130603790	0.318113252410
$\delta = 6$	1.40922594070	3.28039193153	2.05482839346	0.261217096578
$\psi_{geo}(N,\delta)$, $d_{geo} \approx 0.7157$, websize 320KB				
Circ. length	Min.	Max.	Avg.	Std. dev.
$\delta = 3$	1.41799902916	2.93465995789	2.20432470083	0.310828573513
$\delta = 4$	1.54156398773	3.33606600761	2.37035997391	0.329438846284
$\delta = 5$	1.88031601906	4.10431504250	2.51430423737	0.370494801277
$\delta = 6$	1.64570999146	3.89323496819	2.70262962818	0.376313686885

Table 6.3: Experimental results, table ψ_{geo}

value. Nevertheless, this selection of nodes methodology suffers from a higher penalty in terms of latency in accordance with the extrapolated results of our evaluation. As it can be inferred from the analysis of the numerical outcomes, and reflected in Figure 6.5, the random selection algorithm exhibits the worst transmission times, regardless of the size of the site or the length of the circuit used. This can be explained by the random nature of this strategy. Indeed, by selecting the nodes at random, the strategy can incur in some problems which affect directly to the latency of a computed circuit, such as a big distance between the involved nodes (in terms of countries, *i.e.*, routers), a network congestion in a part of the circuit, or a selection of nodes with limited computational resources, among others. It is clear that all these drawbacks are hidden to the strategy and explain the obtained results. Moreover, all these problems are reflected in the standard deviation of the measurements, which is the higher one compared with the other alternatives.

6.3.4 Geographical selection of nodes strategy evaluation

The evaluation of the geographical selection of nodes strategy has been performed by fixing the country and taking into consideration the node distribution detailed in Table 6.1. United States was selected in accordance to the country where the client node resides. Therefore, we can calculate the anonymity degree for this strategy by recalling its related expression introduced in Chapter 5, Section 5.2.2:

$$d_{geo} = \frac{log_2(m)}{log_2(n)} = \frac{log_2(27)}{log_2(100)} \approx 0.7157$$

As we can observe, the degree of anonymity has dropped significantly when we compare it with the

$\psi_{bw}(N,\delta)$, $d_{bw} \approx 0.9009$, websize 50KB				
Circ. length	Min.	Max.	Avg.	Std. dev.
$\delta = 3$	0.964261054993	5.12318110466	1.86709306002	0.789060168081
$\delta = 4$	1.078310012820	5.41474699974	2.36407416582	0.859666129425
$\delta = 5$	1.060457944870	6.92380499840	2.63418945789	1.128347022810
$\delta = 6$	1.278292894360	12.7536408901	3.03272451162	1.882337407440
$\psi_{bw}(N,\delta)$, $d_{bw} \approx 0.9009$, websize 150KB				
Circ. length	Min.	Max.	Avg.	Std. dev.
$\delta = 3$	1.26475811005	7.09091401100	2.28234255314	0.765374484505
$\delta = 4$	1.23797798157	6.80870413780	2.91089500189	0.947719280103
$\delta = 5$	1.45632719994	12.6443610191	2.97445464373	1.431690789930
$\delta = 6$	1.27809882164	12.7246098518	3.19875429869	1.666334473980
$\psi_{bw}(N,\delta)$, $d_{bw} \approx 0.9009$, websize 320KB				
Circ. length	Min.	Max.	Avg.	Std. dev.
$\delta = 3$	1.49932813644	12.9250459671	3.29500451326	1.79104222251
$\delta = 4$	1.52931094170	13.7227480412	3.70603173733	1.90767488259
$\delta = 5$	1.66296601295	17.3828690052	4.07738301039	2.18405609668
$\delta = 6$	2.04065585136	20.1761889458	4.32070047140	2.68160673888

Table 6.4: Experimental results, table ψ_{bw}

results of the other strategies. However, sacrificing a certain level of anonymity incurs in a drastic fall of the latency needed to download a Web page, as it can be noticed if we compare Figures 6.5a, 6.5b and 6.5c. In fact, this selection of nodes methodology provides the best performance in terms of the time required to download a Web page among the other alternatives. It is also interesting to remark the fact that the standard deviation of the time measured in this method remains nearly constant regardless of the circuit length and the size of the Web page. This seems reasonable since the more geographically near are the nodes, the less random interferences affect to the whole latency. We can understand this if we think in terms of the number of networks elements (*i.e.*, routers, switches, etc.) involved in the TCP/IP routing process between every pair of nodes. Thus, a pair of nodes which belong to the same country will be interconnected through less network elements compared to two nodes which belong to different countries and, as a consequence, the latency will be more stable along time. This can be an interesting fact, since the penalty introduced by the use of Tor affects less to the psychological perception of the user when browsing the Web [88]. Nevertheless, the anonymity degree of this strategy is strongly tied to the fixed country, since —as we pointed out in Theorem 5 (*cf.* Chapter 5, Section 5.2.2)— the less nodes belonging to the country, the less anonymity degree is provided.

6.3.5 Bandwidth selection of nodes strategy evaluation

The anonymity degree of the bandwidth selection of nodes strategy has been computed empirically according to its associated formula (*cf.* Chapter 5, Section 5.2.3 for details). In particular, the `torspd.py` application was in charge of obtaining the bandwidth of every node of our private Tor network and of

$\psi_{grp}(N,\delta)$, $d_{grp} \approx 0.9568$, websize 50KB				
Circ. length	Min.	Max.	Avg.	Std. dev.
$\delta = 3$	0.935021877289	3.61296200752	1.59488223791	0.545028374794
$\delta = 4$	0.998504877090	3.74897003174	1.77225045919	0.548956074123
$\delta = 5$	1.195134162900	4.21774697304	2.02931211710	0.576776679346
$\delta = 6$	1.267808914180	3.35924196243	2.18245174408	0.502899662482
$\psi_{grp}(N,\delta)$, $d_{grp} \approx 0.9568$, websize 150KB				
Circ. length	Min.	Max.	Avg.	Std. dev.
$\delta = 3$	1.112107038500	5.53429508209	2.04227621531	0.790901626275
$\delta = 4$	1.290552854540	5.68215894699	2.66674958944	0.944641197284
$\delta = 5$	1.163586854930	7.41387891769	2.68937173843	0.917799034111
$\delta = 6$	1.550453186040	5.40707683563	3.00299987316	0.935654846647
$\psi_{grp}(N,\delta)$, $d_{grp} \approx 0.9568$, websize 320KB				
Circ. length	Min.	Max.	Avg.	Std. dev.
$\delta = 3$	1.502956867220	7.29033994675	2.51847231626	1.009688576850
$\delta = 4$	1.498482227330	6.52234792709	3.22330027342	1.061893260420
$\delta = 5$	1.734797000890	6.73247194290	3.31047295094	0.940285391625
$\delta = 6$	1.689666986470	7.89933013916	3.46063615084	1.094579395080

Table 6.5: Experimental results, table ψ_{grp}

calculating the anonymity degree. Thus, the anonymity degree when the evaluation of this strategy was performed was approximately 0.9009. It is important to highlight that, in spite of the fixed bandwidth specified in the configuration, the bandwidth of every onion router is estimated periodically by the Tor software running at every node, and provided later to `torspd.py` through the Directory Servers. Indeed, if we think that the established bandwidth of a node through its configuration does not necessarily correspond to the real value, then the anonymity degree can change in time in comparison to the previous strategies.

From the viewpoint of the latency results, we can observe how the bandwidth selection of nodes strategy improves the values respect to the random strategy by sacrificing some degree of anonymity. However, it does not achieve the transmission times of the geographical methodology. The reason for that is because this strategy does not take into account important networking aspects, such as network congestion, number of routers, etc., that also impact the transmission times. Therefore, it is fairly reasonable that this methodology is more susceptible to networking problems, resulting in an increase of the eventual transmission time results. This is also corroborated by the standard deviation results, noting the lack of stability of the results. In fact, the transmission times increase as the size of the Web page or the length of the circuit also increase.

6.3.6 Graph of latencies strategy evaluation

The experimental evaluation of our proposal has been performed after the establishment of the parameters of its related algorithms. In particular, they were $\Delta t = 5$, $m = 3$, $k = 300$ and $max\_iter = 5$. Fur-

thermore, the *Latency Computation Process* was launched two hours before the execution of `webspd.py`, leading to an analytical graph with a set of more than 3,000 edges, and which represents a density value of, approximately, 0.67. At this moment, the `torspd.py` estimated the degree of anonymity in accordance to the formula presented in Section 6.2.3. Since such equation depends on the length of the circuit, the anonymity degree was estimated for lengths 3, 4, 5 and 6, giving the results of 0.9987, 0.9984, 0.9982 and 0.9981, respectively. As occurs with the previous strategy, the degree of anonymity is dynamic over time, and in this case depends on the connectivity of the analytical graph. Nevertheless, the anonymity degree was not estimated again during the evaluation tests.

Function c_t was implemented by means of the construction of random circuits of length m. Such circuits are not used as anonymous channels for Web transmissions, but to estimate the latencies of the edges. This is possible since during the construction of a circuit, every time a new node is added to the circuit, the *Latency Computation Process* is notified. Hence, it is easy to determine the latency of an edge by subtracting the time instants of two nodes added consecutively to a certain circuit. Regarding this *modus operandi* of measuring the latencies, it is interesting to highlight two aspects. The first one is that it meets the restriction of estimating the latencies secretly; and the second one is that it not only measures the latencies in relation the network solely, but also takes into consideration delays motivated by the status of the nodes or its resources limitations. This way, our proposal models indirectly some negative issues which the other strategies do not reflect, leading to an improvement of the transmission times as the obtained results evidence.

By comparing the results of the previous strategies with the current one, we can observe how our new proposal exhibits a better trade-off between degree of anonymity and transmission latency. Particularly, from the perspective of the transmission times, our proposal is quite close to those from the geographical selection strategy, while it provides a higher degree of anonymity. Indeed, if we compare our strategy from the anonymity point of view, we can observe that only the random selection of nodes criterion overcomes our new strategy, but, as already mentioned, by sacrificing considerably the transmission time performance.

6.4 Related work

The use of entropy-based metrics to measure the anonymity degree of infrastructures like Tor was simultaneously established by Diaz *et al.* [50] and Serjantov and Danezis [122]. Since then, several other authors have proposed alternative measures [71]. Examples include the use of the min entropy by Shmatikov and Wang in [128], and the Renyi entropy by Clauß and Schiffner in [45]. Other examples include the use of combinatorial measures by Edman *et al.* [58], later improved by Troncoso *et al.* in [145]. Snader and Borisov proposed in [133] the use of the Gini coefficient, as a way to measure inequalities in the circuit selection process of Tor. Murdoch and Watson propose in [101] to assess the bandwidth available to the adversary, and its effects to degrade the security of several path selection techniques.

With regard to literature on selection algorithms, as a way to improve the anonymity degree while also increasing performance, several strategies have been reported. Examples include the use of reputation-based strategies [17], opportunistic weighted network heuristics [133, 134], game theory [159], and system awareness [59]. Compared to those previous efforts, whose goal mainly aims

at reducing overhead via bandwidth measurements while addressing the classical threat model of Tor [140], our approach takes advantage of latency measurements, in order to best balance anonymity and performance. Indeed, given that bandwidth is simply self-reported on Tor, regular nodes may be mislead and their security compromised if we allow nodes from using fraudulent bandwidth reports during the construction of Tor circuits [17, 62].

The use of latency-based measurements for path selection on anonymous infrastructures has been previously reported in the literature. In [126], Sherr *et al.* propose a link-based path selection strategy for onion routing, whose main criterion relies, in addition to bandwidth measures, on network link characteristics such as latency, jitter, and loss rates. This way, false perception of nodes with high bandwidth capacities is avoided, given that low-latency nodes are now discovered rather than self-advertised. Similarly, Panchenko and Renner [111] propose in their work to complement bandwidth measurements with round trip time during the construction of Tor circuits. Their work is complemented by practical evaluations over the real Tor network and demonstrate the improvement of performance that such latency-based strategies achieve. Finally, Wang *et al.* [150, 151] propose the use of latency in order to detect and prevent congested nodes, so that nodes using the Tor infrastructure avoid routing their traffic over congested paths. In contrast to these proposals, our work aims at providing a defence mechanism. Our latency-based approach is considered from a node-centred perspective, rather than a network-based property used to balance transmission delays. This way, adversarial nodes are prevented from increasing their chances of relying traffic by simply presenting themselves as low-latency nodes, while guaranteeing an optimal propagation rate by the remainder nodes of the system.

6.5 Conclusion

We addressed in this chapter the influence of circuit construction strategies on network latency and the anonymity degree of the Tor anonymity infrastructure. We presented the construction of a new circuit selection algorithm that considerably reduces the success probability of linking attacks while providing enough performance for low-latency services. This proposal is based on the concept of *latency graphs*. Together with our approach, we evaluated three classical strategies, with respect to their de-anonymisation risk and latency, and regarding its performance for anonymising Internet traffic. Our experimental results, conducted on a real-world Tor deployment over PlanetLab, confirm the validity of the new strategy, and shows that it outperforms the classical ones.

7

Scalable and single-pass key agreement protocol for Tor circuit establishment

> *I understood the importance in principle of public key cryptography but it's all moved much faster than I expected. I did not expect it to be a mainstay of advanced communications technology.*
>
> WHITFIELD DIFFIE

In Chapter 5 and 6 we have shown how the Tor node selection algorithms can exercise an important influence on the network latency experienced by the users. This assertion can be applied not only to the latency exhibited by an established circuit, but also to the process of a construction of a new one. In spite of this, there is an additional degree of improvement from the point of view of the latency and the process of building a circuit. In this case, this improvement emerges from the use of cryptographic schemes that reduce the number of exchanged messages between the client and the nodes.

As we stated in Chapter 2, Tor uses a technique called *telescoping* where the circuit is built incrementally, negotiating a symmetric key with each onion routing on the circuit, one hop at a time. Such protocol is known as The Authentication Protocol (TAP). Although other proposals have improved the efficiency of the telescoping method used in TAP, the main drawback of these solutions is the degree of interchanged messages. Indeed, to build a circuit composed by n routers adopting a telescopic strategy, it is required the exchange $n(n+1)$ symmetrically encrypted messages, which implies a complexity of $\mathcal{O}(n^2)$. Some published papers have improved this by means of applying different methods such

as identity-based cryptography, Certificateless Public Key Encryption, or Diffie-Hellman Chains among others. All of these schemes relies on enhancing the establishment of the shared keys between the client and each node of a circuit. However, these proposals introduce another problems such as scalability or lack of security properties. In this chapter, we work towards a protocol that improves the establishment of shared keys in the circuit construction, reducing the degree of interchanged messages in comparison to other previous works.

7.1 Key establishment protocols and related work

As we have depicted in Chapter 2, the establishment of shared keys between the clients and the Tor nodes is one overriding need in order to ensure the anonymity of the communications. In this section, we review some concepts regarding the establishment of keys and the related work in such context.

7.1.1 Key establishment protocols

According to Law *et al.* [90] we can define the concept of *key establishment* as the process by which two (or more) entities establish a shared secret key over an insecure network controlled by an adversary. The resulting key is then used to achieve some cryptographic goal such as anonymity, confidentiality or data integrity among others. We can consider that there are two kinds of key establishment protocols: *key transport protocols* in which a key is created by one entity and securely sent to the second party, and *key agreement protocols* in which both entities contribute with some information to establish the shared secret key.

Let A and B be two honest entities that legitimate execute a protocol, we said that a key agreement protocol provides *implicit key authentication* (of B to A) if entity A is assured that no other entity aside from a specifically identified second entity B can possibly obtain the established secret key. It is important to remark that implicit key authentication does not necessarily guarantee that A is assured of B possesses the key. A key agreement protocol which provides implicit key authentication to both participating entities is called an *authenticated key agreement* (AK) protocol. In the context of the Tor network, we are specially interested in the two-party authenticated key agreement protocols, where one entity is the client and the other a Tor node.

In the context of the key agreement protocols, we can also consider another taxonomy. This classification is based on the participation degree of the entities and the exchanged messages during the protocol [106]. It is comprised by three different families of protocols, namely, *two-pass* (or *one-round*), *single-pass* (also known as *one-flow* or *non-interactive*), and *full non-interactive*. In the *two-pass* family, both parties require to transmit information to each other in order to establish a shared key. In the *single-pass* family, just only one entity is required to transmit information to the other one. Finally, the *full non-interactive* family does not require the transmission of information between the two parties.

7.1.2 Related work

Taking into consideration the taxonomy presented in the previous section, we can classify the proposed key agreement protocols concerning to the Tor network (see Figure 7.1). Following, we briefly describe

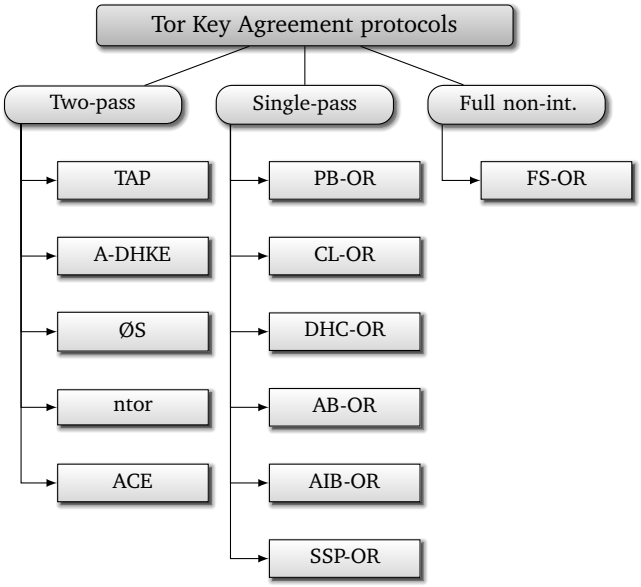


Figure 7.1: Classification of the Tor key agreement protocols

such protocols according the family they belong to. We encourage the reader to consult the original sources for further details.

Two-pass key agreement protocols

Following, we describe the current protocols that belong to the family of two-pass schemes:

- **TAP:** The Tor Authentication Protocol (TAP) was proposed by Dingledine *et al.* [52]. It basically performs a Diffie-Hellman key exchange, where the ephemeral key g^x of the client is encrypted under the public key of the server before is sent. In this way, the encryption guarantees that the server is authenticated. When the server receives the encrypted message, it is decrypted and the ephemeral key g^x is recovered. Then, it calculates its ephemeral key g^y which is sent back to the client. Thus, both the client and the server can compute the established session key $(g^x)^y = (g^y)^x$. TAP was formally proven secure by Goldberg [66], and is one of the two supported protocols in the current implementation of Tor.

- **A-DHKE:** Proposed by Shoup [129] it is also based on the Diffie-Hellman protocol. First, the client computes its ephemeral key g^x which is sent to the server. Upon the reception of g^x by the server, it computes g^y. Then, it obtains the digital signature of the concatenation of the two ephemeral keys using its private key. The ephemeral key g^y along with the signature is sent to the client. Thus, the client can compute the shared key $(g^x)^y = (g^y)^x$ after the verification of the signature by using the public key of the server. Shoup proven that the protocol is secure against adaptive user corruptions, and adaptive user instance corruptions, under the Decisional Diffie-Hellman assumption and secure signatures.

- **ØS:** Øverlier and Syverson presented this solution [109] as the fourth version of a set of more efficient protocols compared to TAP. Again, it is supported by the Diffie-Hellman key exchange,

and a long-term public key of the server g^b. It operates in the following way. First, the client sends an ephemeral key g^x to the server. When the server receives g^x, it obtains its ephemeral key g^y and computes the shared key as $(g^x)^{b+y}$. Then, g^y is send back to the client, which computes $(g^b g^y)^x = (g^x)^{b+y}$. Lamentably, Goldberg, Stebila and Ustaoglu discovered a Man-In-the-Middle attack which was published in [67]. The attack is performed as follows. The attacker intercepts the ephemeral key of the client g^x, computes g^y, and responds with $g^y/g^b = g^{y-b}$. As a result, the client obtains the shared key $(g^b g^{y-b})^x = g^{yx}$, which can be also calculated by the attacker as $(g^x)^y$.

- ntor**:** This key-exchange protocol was published by Goldberg, Stebila and Ustaoglu as a corrected version [67] of the ØS protocol. They observed that in order to fix the attack, it is suffice to decouple the terms xy and xb in the established shared key g^{xy+xb}. In order to accomplish this, they proposed to use a hash function applied to the concatenated terms g^{xy} and g^{xb}, that is $H(g^{xy}, g^{xb})$. In such a way, the protocol is performed as follows. The client computes and sends the ephemeral key g^x to the server. When the server receives the g^x key, it computes its ephemeral key g^y, and obtains the session key $H((g^x)^y, (g^x)^b)$. Thereafter, the g^y value is sent to the client, which computes the key as $H((g^y)^x, (g^b)^x)$. The authors also specified a formal model called *one-way Authenticated Key Exchange* (1W-AKE) based on the eCK model [89]. Their model provides a framework to prove the security properties of this protocol. The ntor was proposed to be implemented in the Tor specification[1], and was included in the *0.2.4.8-alpha* branch (*cf.* the `UseNTorHandshake` configuration parameter).

- **ACE:** The ACE protocol was published by Backes, Kate and Mohammadi in [14]. They improved the computational efficiency and security compared to the previous proposals. Also, the protocol is formally proven secure under the 1W-AKE model. This key-exchange protocol works as follow. First, the client choose an ephemeral key pair (g_1^x, g_2^x). Following, the ephemeral key pair is sent to the server which responds with another ephemeral key g^y. Then, the client computes the shared key as $(g^b)^{x_1}(g^y)^{x_2} = g^{bx_1+yx_2}$ where g^b is the public key of the server. Analogously, the server computes the session key as $(g^{x_1})^b(g^{x_2})^y = g^{bx_1+yx_2}$. The ACE protocol was included in the list of the Tor proposals[2], however, it is not implemented yet.

Single-pass key agreement protocols

The protocols that are included in the family of single-pass schemes are described below:

- **PB-OR:** This protocol, published by Kate *et al.* in [82, 83] is indeed inspired on the proposal of Boneh-Franklin identity-based encryption setup [24]. In this scheme, a trusted party called Private Key Generator (PKG) generates private keys (d_i) for clients using their public identities (ID_i) and a master secret s. A client that has an identity ID_i receives the private key $d_i = sQ_i \in \mathbb{G}$, where $Q_i = H(\mathsf{ID}_i)$, $H : \{0,1\}^* \rightarrow \mathbb{G}^*$ is a cryptographic hash function, and $\mathbb{G}^*$ denotes the group $\mathbb{G}$ except the identity. Sakai *et al.* observed in [119] that with such setup, any two clients that belong to the same PKG can establish a shared key using only the public identities and their own private

[1] https://gitweb.torproject.org/torspec.git/tree/proposals/216-ntor-handshake.txt
[2] https://gitweb.torproject.org/torspec.git/tree/proposals/223-ace-handshake.txta

keys. Dupon and Enge proved in [56] that this protocol is secure in the random oracle model and the BDH assumption. Kate *et al.* adapted this protocol for the Tor context as follows. Suppose that Alice with the key pair (ID_A, d_A) wants to establish anonymously a shared key with Bob who has the key pair (ID_B, d_B). Alice generates a random number $r_A \xleftarrow{\$} \mathbb{Z}_n^*$, creates a pseudonym $P_A = r_A Q_A$ and its corresponding private key $r_A d_A = sP_A$. Alice computes the shared key as $K_{A,B} = \hat{e}(sP_A, Q_B) = \hat{e}(Q_A, Q_B)^{sr_A}$ and sends P_A to Bob. Bob, using P_A and his private key d_B, gets the session key $K_{A,B} = \hat{e}(P_A, d_B) = \hat{e}(Q_A, Q_B)^{sr_A}$. In order to achieve scalability, authors propose a distributed PKG where a master key is generated in a completely distributed way by means of Shamir secret sharing scheme [123]. In this way, only a subset of the total PKGs must be online in order for a client to retrieve his private key. Also, and with the aim of achieving the property of *Forward Secrecy*, authors propose to change the keys after two interval of times: one related to the private keys of the nodes of Tor, and other with the master key of the PKG. The paper also proves formally that the protocol has the the following properties *Cryptographic unlinkability*, *Integrity and correctness*, and *Key secrecy*.

- **CL-OR:** Catalano *et al.* propose the use of certificateless encryption as the basis to establish a shared key [37, 38]. Certificateless encryption retains the good characteristics of the identity-based encryption while overcomes its deficiencies: the trusted party KGC cannot decrypt ciphertexts encrypted with the established key, and the public keys do not need to be certified. They also prove that its protocol is secure under the Strong Diffie-Hellman Assumption in the random oracle model. Their protocol is divided in four phases. (1) *Protocol setup*: the KGC chooses a group $\mathbb{G}$ of primer order n, a random generator $g \in \mathbb{G}$ and two hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_n$ and $H_2 : \mathbb{Z}_n \times \mathbb{Z}_n \to \{0,1\}^l$. Then it obtains a random value $x \xleftarrow{\$} \mathbb{Z}_n$ and computes $y = g^x$. Finally it publishes $MPK = \langle n, \mathbb{G}, g, y, H_1, H_2 \rangle$ as the public parameters and keeps $MSK = x$ secret. (2) *Partial Secret Key Extraction*: after the verification of the identity ID of a client, the KGC picks a random $k \xleftarrow{\$} \mathbb{Z}_n$ and sets $r = g^k$. At that moment, it calculates $s = k + H_1(\mathsf{ID}, r)x$, and provides the partial secret key to the user ID as $d_{\mathsf{ID}} = (r, s)$. (3) *User's key generation*: once a user ID has its partial secret key d_{ID}, it gets at random $t \xleftarrow{\$} \mathbb{Z}_n$ and sets $u = g^t$. Then it defines its public key as $\mathsf{pk}_{\mathsf{ID}} = (r, u)$ and its secret key as $\mathsf{sk}_{\mathsf{ID}} = (s, t)$. (4) *A protocol session*: when the user U wants to establish a session key with B, it selects at random $w \xleftarrow{\$} \mathbb{Z}_n$ and defines its pseudonym as $P_U = g^w$. Following, it obtains the public key of B as $\mathsf{pk}_B = (r_B, u_B)$ and gets the shared key as $K = H_2(z_1, z_2)$ where $z_1 = (r_B y^{H_1(B, r_B)})^w$ and $z_2 = u_B^w$. Afterwards, U sends P_U to B, who gets the same shared key by calculating $z_1 = P_U^{s_B}$ and $z_2 = P_U^{t_B}$.

- **DHC-OR:** The DHC-OR was proposed by Peng in [112] and considers the use of Diffie-Hellman chains with the aim of greatly saving computation and communications. The protocol works in a cyclic subgroup $\mathbb{G}$ with order q in $\mathbb{Z}_p^*$, where q is a factor of $p - 1$, and being p and q large primes. Also, a generator g of the subgroup $\mathbb{G}$ is considered. Following, each router OR_i chooses its private key $x_i \xleftarrow{\$} \mathbb{Z}_q$, and computes its corresponding public key $y_i = g^{x_i} \bmod p$. Then, the sender computes the shared keys as follows. First, it gets $s_1 \xleftarrow{\$} \mathbb{Z}_p$. After that, for each router in the circuit OR_i, for $i = 2, 3, ..., n + 1$, the sender calculates $s_i = s_{i-1} + k_{i-1} \bmod q$ and the shared key $k_i = y_i^{s_i}$. After that, the sender computes the base $b_1 = g^{s_1}$ which is sent, among other

information, to the first router OR_1. Subsequently, each router OR_i that receives a new message from the previous router (or the sender) computes its shared key as $k_i = b_i^{x_i}$. Then, it calculates the new base as $b_{i+1} = b_i g^{k_i} \bmod p$, which is sent to OR_{i+1}. It is important to remark that, each message sent by the sender or a router to the next node in the circuit, not only includes the base b_i, but also the message and the next OR in an onion form. In spite of the efficiency of this protocol from the perspective of computation, it is worth noting that it suffers some security deficiencies like the absence of *forward secrecy*.

- **AB-OR:** The AB-OR protocol makes use of the Ciphertext Policy Attribute Based Encryption (CP-ABE) based on bilinear pairings to improve other routing schemes. At the same time, it achieves failure tolerance properties from the point of view of the Tor nodes. It was proposed by Nishant *et al.* in [55]. The core of the protocol is based on two main phases: *Keygen* and *Circuit_construction*. The *KeyGen* is executed by the PKG to create the private keys for the sender, receiver and Tor nodes. In this phase, the PKG selects $r \xleftarrow{\$} \mathbb{Z}_p$ and $r_j \xleftarrow{\$} \mathbb{Z}_p$. Then it computes the private key as $\mathsf{sk}_{\mathsf{ID}} = (D = g^{(y+r)/\beta}, D1 = g^r H(\mathsf{ID})^{r_j}, D_1' = g^{r_j})$. The *Circuit_construction* is executed by the sender. It takes a message M, constructs a policy $W = \{R\}$ and computes $CT = Encrypt(MPK, M, W')$. Then, constructs a new policy $W = \{OR_1$ or OR_2 or ... or OR_n or $R\}$ and computes $CT = Encrypt(MPK, CT', W)$. The routine $Encrypt$ is the same that the presented in [20].

- **AIB-OR:** The AIB-OR solution published by Wang *et al* in [149] is, according to their authors, an improved version of the PB-OR proposed by Kate *et al.* in [82]. The protocol works as follows. Suppose that Alice wants to perform a session key agreement with Bob. Alice chooses $r_A \xleftarrow{\$} \mathbb{Z}_q^*$ and computes her pseudonym as $PN_A = r_A P$ and its corresponding private key $\mathsf{sk}_{PN_A} = r_A P_{pub} = sPN_A$. Then, Alice gets her pseudonym certificate $\langle PN_A, \sigma_A \rangle$. In order to get σ_A Alice proceeds in this way. First, it gets $r_X \xleftarrow{\$} \mathbb{Z}_q^*$ and generates a masked pseudonym by computing $PN_A' = r_X H(PN_A)$. Then Alice sends PN_A' to the PKG. The PKG computes $\sigma_A' = sPN_A'$ and sends the signature σ_A' to Alice. Upon the reception of σ_A', Alice verifies σ_A' by testing $\hat{e}(\sigma_A', P) = \hat{e}(PN_A', P_{pub})$. If the expression holds, Alice gets $\sigma_A = r_X^{-1}\sigma_A' = sH(PN_A)$. Then, Alices computes the session key $K_{A,B} = e(\mathsf{sk}_{PN_A}, Q_B)$ and sends her pseudonym certificate $\langle PN_A, \sigma_A \rangle$ to Bob. Bob verifies Alice's pseudonym by checking $\hat{e}(\sigma_A, P) = \hat{e}(H(PN_A), P_{pub})$. If the equation holds, Bob gets the corresponding session key $K_{A,B} = \hat{e}(PN_A, d_B)$.

Full non-interactive key agreement protocols

Currently, in the context of full non-interactive protocols we have just found one proposal which is called FS-OR:

- **FS-OR:** Catalano *et al.* present in [39] the first full-non-interactive protocol for the establishment of Tor circuits. They first propose a generic construction based on five algorithms. (1) $\mathsf{Setup}(1^k, T)$: given a security parameter k and a number of maximum periods T, it outputs a public key MPK and a master secret key MSK. (2) $\mathsf{KeyGen}(MSK, \mathsf{ID}, t)$: given the master secret key MSK, a time period t, the algorithm produce a private key $sk_{\mathsf{ID},t}$ related to the identity ID. (3) $\mathsf{KeyUpdate}(sk_{\mathsf{ID},t})$: on input $sk_{\mathsf{ID},t}$, the algorithm outputs a new key for time period $t+1$. (4)

Encap(MPK, ID, t): given the master public key, an identity ID and a time period t, the algorithm outputs a ciphertext C and a session key K. (5) Decap$(sk_{\mathsf{ID},t}, C)$: using the secret key associated to the identity ID and the time period t, the algorithm recovers the session key K from the ciphertext C. Afterwards, authors define an instantiation of their proposal based on the HIBE of Boneh, Boyen and Goh [23] and the generic construction of Canetti-Halevi-Katz [30]. From a pure standpoint of key agreement, the idea is that senders encrypt messages using the public key of the destination and a time period t. Then, at each time period and in order to provide forward secrecy, receivers update their secret key from $sk_{\mathsf{ID},t}$ to $sk_{\mathsf{ID},t+1}$ and deletes sk_{ID_t}. This update is performed as a one-way process (it is hard to reverse the process).

7.2 Proposed key agreement protocol

We expose in this section a new scalable scheme for the construction of Tor circuits in a single pass based on bilinear pairings, and called SSP-OR. In order to achieve such scalability, the protocol allows that the participant entities use several different KGCs, improving previous approaches that relies in just only one. Our protocol is based on a modification of the scheme proposed by Chen and Kudla in [41]. Also, it is formally proven the security properties of our proposal according to the model suggested by Kate *et al.* in [83].

7.2.1 Single-pass authenticated key agreement protocol

Our single-pass authenticated key establishment protocol for the construction of Tor circuits is comprised by six different algorithms, namely Setup, SKExtract, EPKGen, PNGen, BlindCert and KeyAgreement that are described below.

Setup: A Key Generator Center KGC runs $\mathcal{G}(1^k)$ to obtain a prime number n, two groups $\mathbb{G}$ (written additively) and $\mathbb{G}_T$ (written multiplicatively) of order n, a generator P of $\mathbb{G}$, and a bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Then, the KGC selects three hash functions: $H_1 : \{0,1\}^* \to \mathbb{G}^*$, $H_2 : \mathbb{G} \to \mathbb{G}^*$, and $H_3 : \mathbb{G} \times \mathbb{G} \to \{0,1\}^l$ being l the length of the key associated to the cryptosystem that will be used with an established key. Also, the KGC chooses $s \xleftarrow{\$} \mathbb{Z}_n^*$ and computes $P_{pub} = sP \in \mathbb{G}$. Following, it sets the Master Secret Key as $MSK = s$ and the Master Public Key as $MPK = \langle q, \mathbb{G}, \mathbb{G}_T, \hat{e}, P, P_{pub}, H_1, H_2, H_3 \rangle$

SKExtract: A user with an identity ID chooses a KGC that computes $Q_{\mathsf{ID}} = H_1(\mathsf{ID})$ and $d_{\mathsf{ID}} = sQ_{\mathsf{ID}}$. Then, the KGC sends to the user the private key d_{ID}. Note that there is no secure channel between the KGC and the user, since publishing d_{ID} has no effect on the security of the protocol.

EPKGen: An entity X selects $r_x \xleftarrow{\$} \mathbb{Z}_n^*$ and outputs $T_X = r_X P$ as its ephemeral public key.

PNGen: An entity X selects $r_x \xleftarrow{\$} \mathbb{Z}_n^*$ and computes $PN_X = r_X P$ as its pseudonym.

BlindCert: An entity X with a pseudonym PN_X chooses $k_X \xleftarrow{\$} \mathbb{Z}_n^*$ and generates a masked pseudonym $PN'_X = k_X H_2(PN_X)$. Then, X sends PN'_X to a KGC. The KGC computes $\sigma'_X = sPN'_X$ and sends

it to X. Upon receiving the signature σ'_X, X verifies it by checking $\hat{e}(\sigma'_X, P) = \hat{e}(PN'_X, P_{pub})$. If the equation holds, X computes $\sigma_X = r_X^{-1}\sigma'_X = sH_2(PN_X)$, and obtains his pseudonym certificate $\langle PN_X, \sigma_X \rangle$. Anyone in possession of $\langle PN_X, \sigma_X \rangle$ can verify its validity by testing if the expression $\hat{e}(\sigma_X, P) = \hat{e}(H_2(PN_X), P_{pub})$ holds. Note that the algorithm is in fact a blind Boneh-Lynn-Shacham (BLS) signature scheme [25], which is proved to be existentially unforgeable under adaptive chosen-message attacks under the computational Diffie-Hellman assumption in the random oracle model.

KeyAgreement: Suppose that Alice wants to establish a session key with Bob. Alice, knows Bob's identity ID_B and wishes to remain anonymous to Bob. Suppose that there are an arbitrary amount of Key Generator Centers available. For the sake of simplicity, we suppose that there are just only two: KGC_1 and KGC_2. We also suppose that they have executed the Setup algorithm. As a result, they obtain their Master Secret Keys s_1 and s_2, and publish their Master Public Keys $\langle q, \mathbb{G}, \mathbb{G}_T, \hat{e}, P, P^1_{pub}, H_1, H_2, H_3 \rangle$ and $\langle q, \mathbb{G}, \hat{e}, P, P^2_{pub}, H_1, H_2, H_3 \rangle$ respectively, where q, $\mathbb{G}$, $\mathbb{G}_T$, $\hat{e}$, P, H_1, H_2 and H_3 are globally agreed beforehand.

Following, Alice and Bob perform the following steps:

- Bob registers with one of the available Key Generator Center, let is say KGC_1, and gets his private key $d_B = s_1 Q_B = s_1 H_1(\mathsf{ID}_B)$ by executing the SKExtract algorithm. Then, he computes his ephemeral public key as $T_B = r_B P$ by performing the EPKGen algorithm. Bob makes public in a directory the values $\langle T_B, \mathsf{KGC}_1 \rangle$, where KGC_1 is a tag that denotes that he has registered against such Key Generator Center.

- Alice chooses one Key Generator Center among the availables. Let us suppose that she selects KGC_2. Then, she executes the PNGen algorithm and gets $PN_A = r_A P$. Following, by performing the BlindCert algorithm, she obtains her corresponding pseudonym certificate $\langle PN_A, \sigma_A \rangle$.

- Alice retrieves from a public directory, and by using the identity ID_B, the information of Bob $\langle T_B, \mathsf{KGC}_1 \rangle$. Then, she computes $K_{AB} = \hat{e}(\sigma_A, T_B)\hat{e}(Q_B, r_A P^2_{pub})$ and obtains the shared key as $H_3(K_{AB}, r_A r_B P)$.

- Alice sends $\langle PN_A, \mathsf{KGC}_2 \rangle$ to Bob, where KGC_2 is a tag that denotes that she has registered against such Key Generator Center.

- Upon the reception of $\langle PN_A, \mathsf{KGC}_2 \rangle$, Bob computes $K_{BA} = \hat{e}(d_B, PN_A)\hat{e}(H_2(PN_A), r_B P^1_{pub})$, and finally gets the shared key as $H_3(K_{BA}, r_A r_B P)$.

Note that if Alice and Bob follow the protocol, they will compute the same shared secret $K_{AB} = K_{BA} = \hat{e}(r_B \sigma_A + r_A d_B, P)$, and the same key $H_3(K_{AB}, r_A r_B P)$.

It is worth noting the utility of the hash function H_3 in the final step performed by Alice and Bob. Its purpose is to avoid the key escrow property, which is not desirable in the context of anonymity. Let us suppose that KGC_1 and KGC_2 collude to obtain the shared secret (or the values s_1 and s_2 are compromised) and the pseudonym PN_A is intercepted. Then it is possible to compute the session key as $\hat{e}(Q_B, PN_A)^{s_2}\hat{e}(H_2(PN_A), T_B)^{s_1}$. Thus, if we apply H_3 to the shared secret $K_{AB} = K_{BA}$ and the

value $r_A r_B P$, the final session key can only be computed by Alice and Bob, since r_A and r_B are only known by them, and the CDH assumption holds given PN_A and T_B.

7.2.2 Security and anonymity

In accordance to the security model proposed by Kate *et al.* in [83], we impose that our key agreement protocol satisfy the same security requirements. We firstly provide a brief description of these properties and later we expose the formal proofs of those properties in relation to our scheme.

- *Unconditional anonymity*: It is impossible to any other participant in the protocol, the KGCs, or any third entity, to learn the identity of an anonymous party.
- *Session key secrecy*: It is infeasible for anyone except the two parties involved in the protocol to determine the established key or, in other words, an attacker should not be able to recover session keys of uncorrupted parties.
- *No impersonation*: It is infeasible for a malicious client of the KGC to impersonate another non-anonymous party.

Unconditional anonymity

In order to verify that our scheme has the unconditional anonymity property, we will prove that if an adversary intercepts the pseudonym of an anonymous client that is constructing a circuit, and the data interchanged previously between such client and a Key Generation Center, neither the adversary nor the Key Generator Centers can link the pseudonym with the interchanged data. To formalise our proof we consider the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:

Setup: The adversary $\mathcal{A}$ publishes the system parameters: a cyclic additive group $\mathbb{G}$ of prime order n (which has bit-length k), a generator P of $\mathbb{G}$, and a hash function $H_2 : \mathbb{G} \to \mathbb{G}^*$. Also, the available Key Generator Centers are initialised in this phase.

Challenge: The adversary $\mathcal{A}$ executes two times the algorithm PNGen, and gets the pseudonyms $PN_0 = r_0 P$ and $PN_1 = r_1 P$. Then, both pseudonyms are provided to the challenger $\mathcal{C}$. Afterwards, the challenger chooses uniformly at random $b \in [0, 1]$. Following, $\mathcal{C}$ runs the BlindCert algorithm against PN_b and a random selected Key Generation Center KGC. As a result, it gets the masked pseudonyms $PN'_b = k_b H_2(PN_b)$ and its corresponding blind signature $\sigma'_b = sPN'_b$. Then, the challenger $\mathcal{C}$ sends to the adversary $\mathcal{A}$ the values PN'_b, σ'_b and the tag KGC.

Guess: Finally, the adversary $\mathcal{A}$ outputs a guess b' for b. The adversary's advantage is defined as:

$$\text{Adv}^{\text{UA}}_{\mathcal{A}}(k) = max\left\{0, \Pr[b' = b] - \frac{1}{2}\right\}$$

We say that the adversary $\mathcal{A}$ wins the game if $\text{Adv}^{\text{UA}}_{\mathcal{A}}(k)$ is non-negligible.

Since $\mathbb{G}$ is a cyclic group of primer order n, $H_2(PN_0)$ and $H_2(PN_1)$ are generators of $\mathbb{G}$. For the uniform random elements $r_0, r_1 \in \mathbb{Z}_n^*$, the masked pseudonyms $PN_0' = r_0 H_2(PN_0)$ and $PN_1' = r_1 H_2(PN_1)$ are also uniform random elements of $\mathbb{G}^*$. Given the masked pseudonyms, and by considering that the Discret Logarithm Problem is hard in $\mathbb{G}$, the adversary $\mathcal{A}$ can not recover the value $H_2(PN_b)$, which means that he can not determine the value of b with a non-negligible advantage. Moreover, and for the very same reason, given the blind signature $\sigma_b' = sPN_b'$, it is not possible to recover the value PN_b'. Consequently, we can affirm that the advantage $\text{Adv}_{\mathcal{A}}^{\text{UA}}(k)$ is negligible.

Session key secrecy

For such property, we adopt a much stronger security restriction: we require that any PPT (Probabilistic, Polynomial-Time) attacker $\mathcal{A}$ should not be able to distinguish a random session key from a real one. Chen and Kudla prove in [41] this security property for a scheme in which ours is inspired. The proof uses the security model proposed by Bellare and Rogaway in [18] and a reduction argument. In particular, they prove that their scheme is provable secure in the BDH assumption and the random oracle model. Since our protocol simply modifies their proposal by using $T_A = PN_A = r_A P$ and $Q_A = H_3(PN_A)$, the proof of security in [41] is easily modified to suit our protocol.

No impersonation

Suppose an adversarial client wishes to impersonate a non-anonymous participant (say, Bob with ID_B) while communicating with an anonymous client (say Alice with pseudonym PN_A). This implies that the adversary would need to compute the shared key $H_3(K, r_A r_B P)$, where $K = K_{AB} = K_{BA}$. This problem can be be dissected in two sub-problems: to compute the shared secret $K = K_{AB} = K_{BA}$, and to obtain $r_A r_B P$.

Given the first sub-problem, let us consider the two different ways to compute the shared secret according to the description provided in Section 7.2:

- In the first case, the shared secret between Alice and Bob is computed by means of the expression $K_{AB} = \hat{e}(\sigma_A, T_B)\hat{e}(Q_B, r_A P_{pub}^2)$. For the first part of the previous equation, that is $\hat{e}(\sigma_A, T_B)$, the adversary knows the value T_B. However, since the value σ_A is only known by Alice, the adversary can not compute the first part of the shared secret K_{AB}. For the second part of the equation, that is $\hat{e}(Q_B, r_A P_{pub}^2)$, we can observe that this is equivalent to solve the BDH problem: to compute $\hat{e}(P, P)^{c r_A s_2}$ given $\langle P, Q_B, PN_A, P_{pub}^2 \rangle$, where $Q_B = H_2(\mathsf{ID}_B) = cP$, $PN_A = r_A P$ and $P_{pub}^2 = s_2 P$. Let us notice that since $\mathbb{G}$ is a cyclic group with P as a generator, there must exist a value $c \in \mathbb{Z}_n^*$ such that $Q_B = H_2(\mathsf{ID}_B) = cP$.

- In the second case, the shared secret is computed as $K_{BA} = \hat{e}(d_B, PN_A)\hat{e}(H_2(PN_A), r_B P_{pub}^1)$. For the first part of the such expression, that is $\hat{e}(d_B, PN_A)$, we consider that the adversary knows the value PN_A and d_B, which implies that it can be obtained. However, for the second part of the equation, that is $\hat{e}(H_2(PN_A), r_B P_{pub}^1)$, we can observe again that this is equivalent to solve the BDH problem given the values $\langle P, PN_A, T_B, P_{pub}^1 \rangle$, where $PN_A = r_A P$, $T_B = r_B P$ and $P_{pub}^1 = s_1 P$.

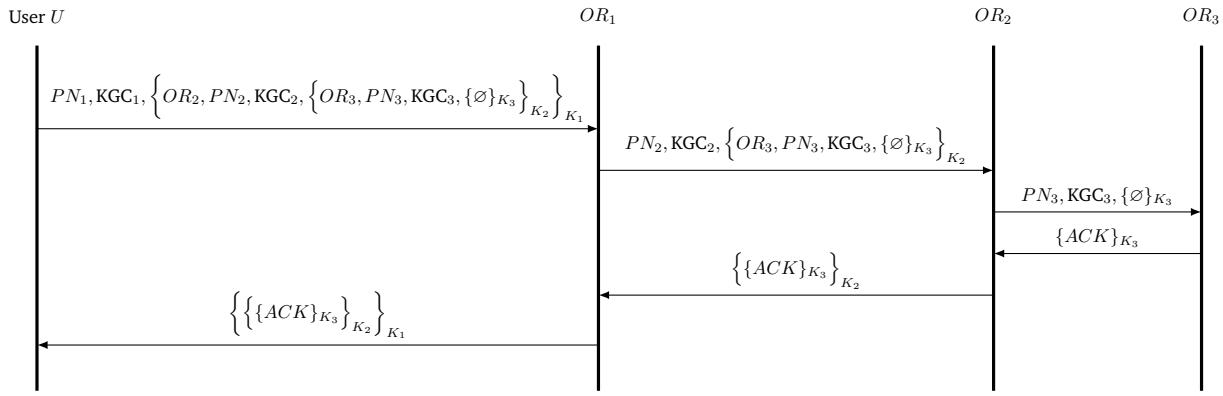


Figure 7.2: Construction of a Tor circuit using our scheme SSP with three ORs

Given the second sub-problem, it is easy to see that this is equivalent to solve the CDH problem as described in Section 2.1.5, and given $\langle P, PN_A, T_B \rangle$.

We can conclude that our scheme has the property of no impersonation by considering the BDH and the CDH assumptions for $\langle n, \mathbb{G}, \mathbb{G}_T, \hat{e} \rangle$ and $\langle n, \mathbb{G} \rangle$ respectively, and the fact that an adversary can not compute σ_A.

7.3 Tor circuit construction and onion routing

We expose in this section how our protocol can be integrated with he construction of Tor circuits in a single pass, and how traffic can be sent through a circuit when it has already been built. We also review in this section the security of the onion routing by using our key agreement scheme.

7.3.1 Tor circuit construction with our scheme

Let $\mathcal{SSP} = (\mathsf{Setup}, \mathsf{SKExtract}, \mathsf{EPKGen}, \mathsf{PNGen}, \mathsf{BlindCert})$ be an instantiation of our scalable single-pass key agreement protocol. Then, we describe our Tor circuit construction protocol based on the following three phases: OrSetup, KeyGeneration and CircuitConstruction. Each of these phases are described below.

OrSetup: In this phase all the KGCs are initialised by executing the Setup algorithm described in Section 7.2. All of them use the parameters q, $\mathbb{G}$, $\mathbb{G}_T$, $\hat{e}$, P, H_1, H_2 and H_3 globally agreed beforehand.

KeyGeneration: All the available onion routers select its own KGC and registers against it. This is performed by running the SKExtract and EPKGen algorithms. Afterwards, they publish their ephemeral keys and KGC tags in a public directory.

CircuitConstruction: A user U retrieves, from a directory server, the list of available onion routers. Then, by using some node selection criterion, he chooses an ordered set of n different onion routers $OR_1, OR_2, ..., OR_n$. These routers will constitute the nodes of the Tor circuit to build. Afterwards, for each router he executes key agreement protocol according to the description given in Section 7.2. That is, for each OR_i ($i \in \{1, ..., n\}$) he selects a KGC_i and executes the PNGen and BlindCert algorithms. As a

result, he obtains the pseudonym certificates $\langle PN_1, \sigma_1, \text{KGC}_1\rangle, ..., \langle PN_n, \sigma_n, \text{KGC}_n\rangle$. Then he computes the shared keys $K_1, K_2, ..., K_n$. By mean of those established keys, he creates an onion as follows:

$$PN_1, \text{KGC}_1, \Big\{OR_2, PN_2, \text{KGC}_2, \big\{...\{OR_n, PN_n, \text{KGC}_n, \{\varnothing\}_{K_n}...\big\}_{K_2}\Big\}_{K_1}$$

where $\{m\}_{K_i}$ denotes that the message m is symmetrically encrypted using the established key K_i. As one can see, each layer of encryption is defined by a triplet $(PN_i, \text{KGC}_i, C_i)$, where PN_i, KGC_i, and C_i are, respectively, the pseudonym, the KGC, and the ciphertext associated to the router OR_i.

Then, the construction of the circuit works as follows (*cf.* Figure 7.2). The user U sends the above onion to the entry router of the circuit. When a router OR_i receives an onion of the form $(PN_i, \text{KGC}_i, C_i)$, it uses the pseudonym PN_i and the public parameters of the router OR_i to recover the shared key K_i. Then, by using such key, the router decrypts the ciphertext C_i and gets the quadruple $(OR_{i+1}, PN_{i+1}, \text{KGC}_{i+1}, C_{i+1})$. From this quadruple, the node constructs and forwards a new message $(PN_{i+1}, \text{KGC}_{i+1}, C_{i+1})$ to the router OR_{i+1}. If an onion router gets $\varnothing$ after the decryption of a ciphertexts, it knows that is the last node in the circuit and sends back a confirmation message $\{ACK\}_{K_n}$ to the previous router. When an onion router receives a confirmation message, it encrypts it with its associated shared key, and sends it back to the previous router in the circuit. This process is repeated until the confirmation message $\{...\{\{ACK\}_{K_n}\}_{K_{n-1}}...\}_{K_1}$ arrives to the user U, who decrypts it by using the keys $K_1, K_2, ..., K_n$. When U recovers the message ACK, he knows that the circuit has been established and that it can be used to route anonymously additional traffic.

7.3.2 Onion routing security

Camenisch and Lysyanskaya established in [29] a security model for onion routing protocols in the Universal Composability (UC) framework. As Kate *et al.* and Catalano *et al.* state in [83] and [37] respectively, such framework is very restrictive since it considers all the possible attacks. Consequently, those schemes that meet the specification of UC becomes inefficient. Following the suggestion of Kate *et al.*, we use their simplified security model for onion routing in order to verify our approach. The properties defined by such model are the following ones [83]:

- *Cryptographic unlinkability*: This property guarantees that, in a circuit with at least one honest node, it is not possible for an attacker to establish a link between a sender and a receiver. It is important to remark that, as it is pointed out in [83], that network-level attacks are not considered in this case. Kate *et al.* proved in [83] that there is a relation between the cryptographic unlinkability notion and the encryption scheme used. If the symmetric cryptosystem adopted for the construction of the onions provides indistinguishability under chosen-plaintext attack (IND-CPA), then the onion routing has the property of cryptographic unlinkability.

- *Correctness and integrity*: An onion routing protocol has the property of *correctness* if a message reaches its destination and its corresponding onion meets the following restrictions (i) it is correctly constructed by a sender, (ii) it is processed by the routers of a circuit in the correct order, and (iii) each router of the circuit executes the protocol according to its specification. Additionally, an onion routing has *integrity* if routers can recognise those onions that are longer than a upper limit.

- *Key secrecy*: An onion routing protocol has *key secrecy* if an attacker that controls all the routers of a circuits except one, can not recover the secret key established between the sender and the honest node.
- *Circuit position secrecy*: An onion routing protocol has this property if when a Tor node that is part of a circuit receives an onion, it is not capable of learning which position it has in the circuit. This restriction is not required for the entry and exit nodes to meet the *circuit position secrecy*.

Following, we analyse these previous properties from the standpoint of our proposal.

Cryptographic unlinkability

As we have stated previously, the cryptographic unlinkability in an onion routing is implied by the use of a symmetric encryption scheme that is IND-CPA secure. Thus, we impose to use with our approach a cryptosystem that satisfies this requirement in order to meet the cryptographic unlinkability property (*e.g.* AES).

Correctness and integrity

Our scheme trivially achieves correctness, otherwise it could not be possible to establish a circuit and route traffic. From the point of view of integrity, let us consider n the upper bound on the number of routers in the circuit, then an onion message containing more than n layers of encryption can be easily detected by any router just inspecting the size of the packet. This can be applied not only for the routing onions, but also for the key agreement protocol presented in Section 7.2.

Key secrecy

It is easy to observe that in our protocol key secrecy directly follows from the security of the key establishment protocol defined in Section 7.2.2. In particular, our restriction is that any PPT attacker $\mathcal{A}$ should not be able to distinguish a random session key from a real one, guaranteeing that our onion routing proposal has the key secrecy property. Therefore, the proof of Chen and Kudla in [41] applies here as well.

Circuit position secrecy

Unfortunately, our protocol does not meet this property since it is based on several re-encryptions, increasing the size of the onion with each new layer. In fact, Camenisch and Lysyanskaya showed in [29] that it is sufficient to analyse the size of the onion to learn the position of a Tor router. In spite of this, we notice that there are several general solution to make the protocol resistant to this attack at the cost of increase the computation and the size of the onion [83, 29, 49, 81].

7.4 Additional security properties and scalability of single-pass schemes

In this section we review additional security properties of our scheme and the scalability that it offers in comparison to other ones. In particular, we consider other single-pass protocols. We must remark that

Property	PB-OR	CL-OR	DHC-OR	AB-OR	AIB-OR	SSP-OR
Based on	IBE	CBE	DHC	IBE	IBE	CBE
Absence of secure channel	✗	N/A	N/A	✗	✓	✓
Absence of Key Escrow	✗	✓	✓	✗	✓	✓
Forward secrecy	✓	✓	✗	✓	✓	✓
KGC Scalability	✗	✓	N/A	✗	✗	✓
Non-inter. key-update by OR	✗	✓	✓	✗	✗	✓
Proven secure under a model	✓	✓	✗	✗	✗	✓

Table 7.1: Comparison table of single-pass key agreement protocols for onion routing

one of the main goals of our proposal is to reduce the time needed for the construction of a Tor circuit while security properties are guaranteed and, at the same time, scalability is preserved. In that sense, single-pass key agreement protocols are better than two-pass schemes from the point of view of the latency exhibited. The reason for that obeys to the fact that the latency penalty in the construction of a circuit is mainly derived from the network delays and the node selection criterion used [33]. This, in conjunction with the fact that the single-pass protocols have a lower degree of interchanged messages, leads to a better performance from the standpoint of the time needed for the construction of a Tor circuit.

In Table 7.1 we have included a comparison between our protocol (SSP-OR) and other single-pass schemes. The analysed properties are the following one:

- *Absence of secure channel*. Some single-pass protocols (*e.g.* PB-OR) need a secure channel between the trusted party and the OR in order to preserve security and forward secrecy. Otherwise, an attacker could compromise an OR with the absence of such secure channel. In particular, in the PB-OR protocol, this channel is used to send the private key of the OR and generated by the PKG. In contrast, our protocol does not require such channel since the private key sent by the KGC to the OR does not allow, in isolation, to compute the session key.

- *Absence of key escrow*. In the context of circuit construction schemes, some of them have the property of key escrow. This means that eventually a third party could recover the encrypted traffic sent by a user. It is important to remark that, in spite of that in some scenarios this is a desirable property, this is not acceptable when we want to preserve the anonymity of users. In this line, our proposal does not have this characteristic. To prevent this, we use ephemeral keys associated to the participants. In case of the client, the algorithms PNGen and BlindCert allow to compute its ephemeral key. From the point of view of the OR, the EPKGen algorithm is the responsible to calculate its ephemeral key. We have also avoided the possibility that two KGC can collude in order to compute the established key between two entities that use each KGC. This is accomplished by means of the use of the hash function H_3 as we have described in Section 7.2.1.

- *Forward secrecy*. The forward secrecy is defined as the property of communication protocols in which the compromise of long-term keys does not expose past session keys. In the context of anonymity this characteristic must be seen as a requirement. A related notion to forward secrecy

is the *eventual forward secrecy*. Eventual forward secrecy is a way to achieve forward secrecy by means of frequently changing the long-term server keys. In this way, if an adversary gains access to the secret key of a router, it may only can compromise the communications related to the validity period of that key. The trivial way to implement eventual forward secrecy is by forcing the routers to compute new keys and their corresponding certificates, and to the users to retrieve again such certificates. In order to solve this problem two main ideas has been proposed and implemented by Tor circuit establishment schemes. The first one uses identity-based encryption [124], where the public keys of the routers are constructed by the concatenation of the router identity and the validity period. The second one, is to use a certificateless encryption, where each participant has an identity ID with a matching secret key —generated by a KGC—, and together with a public/secret key pair that do not need to be certified. Our proposal uses a certificateless approach to achieve eventual forward secrecy. In our case, there is a link between the identity of the client and the public/secret key pair. We use a pseudonym $PN = rP$ as the public key, and r as the secret key. At the same time, PN is considered the identity of a client which is mapped through the hash function H_3 to $\mathbb{G}$. From the side of the routers, they generate a public/secret key by means of the EPKGen algorithm. This strategy allows the routers to compute a new public/secret key without the intervention of the KGC. Unfortunately, the client needs the participation of the KGC (*cf.* PNGen and BlindCert algorithms in Section 7.2).

- *KGC scalability*. Some key agreement protocols used for the construction of Tor circuits are subject to the presence of a trusted party. Such trusted party is involved —in some manner— in the computation of the private keys of the clients or the ORs. All the protocols that we have analysed consider a common trusted party for all the participants, which introduce a scalability problem. This scalability problem can be considered not only from the viewpoint of performance, but also from the perspective of security. This could mean that, if such trusted party is compromised, the overall network would be affected. Also, if a protocol relies in just only one trusted party, a DDoS attack against it would affect to all the users that make use of the infrastructure. In that sense, our protocol is the first one that has considered the possibility to use different trusted parties (KGCs) at the same time, and by different parties.

- *Non-interaction for key-update by OR*. Several onion routing protocols require an interaction between the ORs and some trusted party in order to update their private keys. Such keys are used for the computation of the session key established between a particular client and a given OR. In our proposal, such interaction can disappear. The reason for that is that we use an ephemeral key that can be computed by each OR independently (*i.e.* the participation of the trusted entity is not needed), and by means of the EPKGen algorithm.

- *Proven secure under a model*. Some key agreement protocols proposed in the literature for the construction of Tor circuit have not been analysed by means of a security model. This could lead to non desirable issues and vulnerabilities. In our case, we have formally proven that our proposal is secure from the point of view of the secure model proposed by Kate *et al.* in [83].

7.5 Computational efficiency

In this section we evaluate the computational efficiency of our protocol in comparison with other schemes. For this purpose, we have selected some representative proposals among all the analysed in Section 7.1. In particular, we have considered the TAP [52], PB-OR [82, 83], CL-OR [37, 38], DHC-OR [112] and ours (SSP-OR). All these protocols mainly differ in the way the symmetric keys are established. Therefore, we discuss analytically the performance when a circuit of length n is build from the perspective of the user and the onion routers. We have also evaluated the aforementioned schemes with security parameters of 80 and 128 bits, and regarding the recommendations of NIST [15] and ECRYPT [57]. We point out that the latter (128 bits) should be considered the one with adequate level of security. The evaluation has been conducted through the implementation of several prototypes based on the C language and the libraries PBC [91] version 0.5.14, and OpenSSL [143] version 1.0.2k. All the prototypes and the libraries have been compiled using GCC version 4.9.4. The environment where the tests have been performed is based on a system with a 2.7Ghz Intel i7-2620M CPU, with 8GB of RAM, and running a Gentoo GNU/Linux with a 4.9.9 kernel and Glibc version 2.23. Following, we analyse the significant operations required for each protocol according to the descriptions given in Section 7.1:

- **TAP:** The TAP protocol requires that the user performs, for each of the n routers, one RSA encryption and two exponentiations for the Diffie-Hellman ephemeral keys. From the point of view of the onion router, it must perform one RSA decryption and two exponentiations for the Diffie-Hellman ephemeral keys. Given a security level of 80 bits we require a 1024-bits RSA modulus and a 1024-bits finite field for Diffie-Hellman. For a security level of 128 bits, the size of the RSA modulus as well as the size of the Diffie-Hellman field must be of 3072-bits. The Tor specification [51] suggests, as a way of optimisation, to use the RSA exponent 65537, and for the Diffie-Hellman the generator 2 with an exponent of 320 bits.

- **PB-OR:** The authors of the PB-OR suggest —in order to improve the computation efficiency— to implement their scheme over a group of points of elliptic curves. In particular, they recommend to use curves of type A. Also, they propose a strategy based on the pre-computation of the master keys and the private keys (*cf.* papers [82, 83] for further details). From the viewpoint of the user, he must perform n exponentiations in $\mathbb{G}$, and n in $\mathbb{G}_T$. In both cases, pre-computation can be used with the aim of speed up the establishment of a circuit. In case of the routers, they must compute one pairing that can be pre-computed by using a fixed parameter.

- **CL-OR:** The authors of CL-OR suggest to implement the protocol under elliptic curves. More specifically, and as a way to improve the performance and reduce the size of the group elements, they advise to use curves of type F. In this scheme, the user must compute three exponentiations, where two of them can be pre-computed on the fixed bases. From the perspective of the routers, they must compute two exponentiations that can not be pre-computed. Regarding the security parameter, we must choose a curve of type F with a group $\mathbb{G}_1$ whose size is 160 bits and 256 bits in order to achieve, respectively, 80 and 128 bits of security.

- **DHC-OR:** The DHC-OR protocol needs, on the one hand, that the user computes $n + 1$ exponentiations that can not be pre-computed. On the other hand, each router must compute two

	Time (ms)	
Operation	80 bits	128 bits
RSA Encryption	0.053	0.109
RSA Decryption	0.759	5.117
Exp. (Tor)	0.325	1.693
Exp. (DH-Chain)	1.206	15.216
Pairing [A]	2.282	10.658
Pairing [A, pp]	0.989	6.443
Exp. in $\mathbb{G}$ [A, pp]	0.381	0.804
Exp. in $\mathbb{G}_T$ [A, pp]	0.058	0.177
Exp. in $\mathbb{G}_1$ [F]	0.597	1.312
Exp. in $\mathbb{G}_1$ [F, pp]	0.078	0.182

Table 7.2: Summary of cost per operation (in ms)

exponentiations that can not be speeded up by means of pre-computation. For security level of 80 and 128 bits, the Diffie-Hellman finite field used must have a size of 1024-bits and 3072-bits respectively.

- **SSP-OR:** In order to improve the performance of our scheme, we also use elliptic curves of type A. Also, our protocol required that the client compute the expression $K_{AB} = \hat{e}(\sigma_A, T_B)\hat{e}(Q_B, r_A P^2_{pub})$ for each router. This means that the user must compute two pairings n times. The first pairing can be pre-computed if we considered that σ_A was obtained previously and that it is a fixed parameter. Unfortunately, the second pairing can not be pre-computed since the related router can not be known until the establishment protocol starts. The router computes the key by means of the expression $K_{BA} = \hat{e}(d_B, PN_A)\hat{e}(H_2(PN_A), r_B P^1_{pub})$, which leads to the execution of one pre-computed pairing and a non-precomputed one.

In Table 7.2 we have summarised the computational cost of the main operations needed by the analysed schemes. We have used the letters *A* and *F* to refer to the curves of that type (in the PBC nomenclature), and *pp* to denote pre-computation. These computational costs allow us to estimate the performance regarding the construction of a Tor circuit of length n. These estimations are presented in Table 7.3 in conjunction with the number of exchanged messages. As can be seen, our proposal does not present the best performance compared to the rest of schemes. We stress out that the reason for that is motivated by the scalability property described in Section 7.4. In order to achieve this characteristic, we require to compute two pairings, which is one of the most expensive operations according to Table 7.2. Notwithstanding, we believe that these computational costs can be assumed in practical scenarios, specially if we consider the benefits of security and scalability that the scheme provides in comparison to other alternatives (*cf.* Table 7.1).

We also note that although our single-pass scheme exhibits a less efficient computational performance compared to other solutions (specially those based on two-pass), the main root of the latency that users may experience during a circuit construction is related to the network delays. This is particularly

	TAP		**PB-OR**		**CL-OR**		**DHC-OR**		**SSP-OR**	
Time (in ms)	Client	OR	Client	OR	Client	OR	Client	OR	Client	OR
80-bits security	$0.70n$	1.38	$0.44n$	0.99	$0.75n$	1.19	$1.21n$+1.21	2.41	$3.27n$	4.56
128-bits security	$3.44n$	4.14	$0.98n$	6.44	$1.68n$	2.62	$15.22n$+15.22	30.43	$17.10n$	17.10
Num. messages	$n(n+1)$		$2n$		$2n$		$2n$		$2n$	

Table 7.3: Comparison of the computational cost for building a circuit of n routers

important if we consider —as we have shown in Chapter 5— the influence that the node selection algorithms have in regard to the network latencies. Thus, a completely random strategy can potentially introduce a higher degree of network latency compared to other criteria (*e.g.* bandwidth, geographical, ...). We also argue that nowadays the delays that can be introduced by the computational algorithms related to the construction of Tor circuits are not comparable with those associated to the networks. This is relevant if we consider the degree of computational processing power that our systems have today. Consequently, we can view our single-pass scheme as an efficient way to reduce the overall time needed for the establishment of a Tor circuit, while preserving security properties and scalability.

7.6 Conclusion

In this chapter we have have reviewed the state of the art of the key agreement protocols for Tor circuit construction. We have classified them under a general taxonomy based on the participation degree of the entities and the exchanged messages during the protocol. This categorisation distinguishes between three different families: *two-pass*, *single-pass* and *full non-interactive*.

We have also presented a new scheme for the establishment of Tor circuits called SSP-OR, which can be seen as another way to reduce the latencies that users perceive during the use of the Tor network. Our proposal is based on an authenticated key agreement protocol which makes use of bilinear pairings. The solution, instead of iteratively build a circuit, constructs the circuit in a single pass (*i.e.* the protocol belongs to the *single-pass* family). Also, and compared to other proposals, it scales better from the perspective of the trusted authorities. Its security properties have been analysed and formally proven under the perspective of the Kate *et al.* model.

8

Conclusions and open problems

> *To know what you know and what you do not know, that is true knowledge.*
>
> CONFUCIUS

This final chapter summarises the main findings with regard to the research conducted, and presents the general conclusions based on the outcomes of the studies presented in this dissertation. Also, an outlook to some future lines of research are provided. For ease of reading, we have divided the chapter in two main section. The first one synthesises the work carried out and the research outcomes along with the contributions. The second one gives an overview of open problems that could be part of the continuity of this research.

8.1 Summary and contributions

The rise of new Internet services, especially those related to the integration of people and physical objects to the net, makes visible the limitations of the DNS protocol. The exchange of data through DNS procedures flows today into hostile networks as clear text. Packets within this exchange can easily be captured by intermediary nodes in the resolution path and eventually disclosed. Privacy issues may thus arise if sensitive data is captured and sold with malicious purposes. In fact, when DNS was designed in the early eighties, it was not intended to guarantee the privacy of people's queries. It was simply conceived as a federated database with information that needed to remain publicly accessible. These privacy deficiencies are indeed a heritage of the vulnerabilities existing in the DNS mechanisms.

Although the attempts to address some of these issues, like the publication and deployment of the DNSSEC extension, some of those weaknesses are still present nowadays.

In this context, a clear example is how botnets exploit such shortcomings. By means of fast-flux networks, botnets protect their C&C and increase its anonymity, availability, load balancing and resiliency to takedown. In Chapter 3 we expose how one the most effective techniques against this type of networks is based on the detection in real-time of the associated fast-flux domains. Among the possible detection strategies, the approaches based on SVM produce reasonable good results. In this sense, the works of Hsu [75] and McGrath [95] provide the first effective solutions. Each of these proposals consider a set of different n-dimensional characteristics for the process of classification. In particular, McGrath suggests the use of characteristics that are intrinsic to the infected systems that act as flux agents. In contrast, Hsu proposes to employ characteristics that are related to the features of the domains. The analysis exposed in Chapter 3 confirms that considering the two spaces of characteristics separately lead to false positives and false negatives. Our contribution in this area proposes to unify both characteristics spaces in a single one in a new SVM. In this way, the deficiencies of one space of characteristics compensate the other one and vice versa. The experimental results obtained demonstrate that our solution substantially improves the previous contributions, reducing the amount of false positives and false negatives.

As we have introduced, another clear example where the DNS protocol presents important deficiencies is in the context of privacy. Regardless of the use of DNS or DNSSEC, the resolution of queries in clear text can be a risk for the privacy of the users. The situation is further aggravated if we consider other technologies such as ENUM. In Chapter 4, and as a way of overcoming these shortcomings, we consider the possibility of using low latency anonymity infrastructures. In particular, and given its widespread use nowadays, we propose the use of Tor. Although the Tor network itself has the ability to perform DNS resolutions natively, it does not currently guarantee the integrity and authenticity of the responses. As a solution, we propose the simultaneous use of Tor with DNSSEC. Alternatively, we propose an approach based on a PIR model and inspired on two previous works surveyed by Zhao *et al.*. In our solution, security deficiencies detected in both contributions of Zhao *et al.* have been addressed. From the experimental results we conclude that, although the two solutions are promising, the PIR techniques provide a shorter resolution times, while increase the bandwidth usage considerably. By contrast, the use of Tor infrastructure gives worst response times but providing lower bandwidth consumption.

From the results of the previous chapter, we discuss in Chapter 5 the reasons why the Tor latency can be affected, and how it can influence the users' privacy. In particular, in such chapter we show that there is a strong dependence between the algorithm for the selection of nodes that will constitute a circuit, the resulting network latency that will be perceived during the use of such circuit, and the degree of anonymity. In this line, one of our key contributions appears in the aforementioned chapter, where we establish a formal model that allows us to obtain an algebraic expression to measure the degree of anonymity of node selection criteria. To achieve this, we assume the Syverson threat model and that for each selection algorithm it is possible to associate a discrete probability distribution. By using the normalised Shannon entropy applied to such distribution we are able to measure the degree of anonymity. This model is useful not only to determine the risk that a specific strategy may involve, but also as a mechanism to compare different algorithms. Moreover, it allows us to study the underlying

properties of the algorithms like, for instance, on what circumstances a strategy achieves its maximum degree of anonymity. In conjunction with experimental measurements of network latencies, a user can make the decision to use a particular selection strategy based on their needs regarding the trade-off between latency and degree of anonymity.

In accordance with what was described in the previous paragraph, in Chapter 6 we propose an algorithm that improves the trade-off between degree of anonymity and latencies in comparison to other classic strategies. This algorithm is based on modelling Tor as a graph, where each graph node is associated with one router of the anonymity network. Each edge of the graph has a tag that corresponds to an estimation of the latency between the nodes that compose it. To estimate these latencies, circuits are established at random which, once built, are discarded for use. The construction of such circuits is merely that of calculating and updating the latencies associated with the edges. By means of this view of the network state it is possible to choose the entry and exit nodes at random, and then compute the least cost path between them. The resulting path provides the rest of the nodes that will be used for the establishment of a new circuit. This circuit will be used for the anonymous communications. Applying the theoretical model described in Chapter 5 by means of using the concept of λ-betweenness, and thanks to the latency measurements in a PlanetLab deployment, we conclude that our strategy constitutes a contribution to this area outperforming other strategies based on random, bandwidth, or geographic selection.

Besides the use of the different strategies for the selection of nodes as a way of reducing latencies, it is also possible to decrease these times by using different cryptographic protocols for the establishment of circuits. Currently, Tor implements a protocol that establishes the shared keys between the client and the nodes of a circuit called TAP. This strategy is encompassed within the family of two-pass protocols, and requires the interchange of information between the client and each node of the future circuit. As an alternative to the protocols of the two-pass family, the single-pass type can improve the times necessary for the construction of Tor circuits. Thus, in the Chapter 7 we analyse the state of the art of the different proposals according to a general taxonomy of families of protocols. In the same chapter we propose a single-pass protocol based on bilinear pairings. This protocol not only reduces the construction times of the circuits in comparison to other approaches but, unlike other proposals, also scales better with respect to the third trusted parties that take part in the process. Also, it is demonstrated that our solution is secure under the model proposed by Kate *et al* in [83].

8.2 Open problems

We review in this section the open problems and future lines of research that emerge naturally as a continuity of the work presented in this dissertation. This is disaggregated in four main lines, namely: *DNS and future secure alternatives*, *botnets and malicious uses of the DNS*, *PIR as a privacy preserving technology for DNS*, and *anonymity infrastructures as a privacy preserving solution*.

8.2.1 DNS and future secure alternatives

Without a doubt, the current DNS protocol presents some deficiencies from the point of view of security and, more specifically, the privacy of the users. Notwithstanding the efforts to overcome these problems

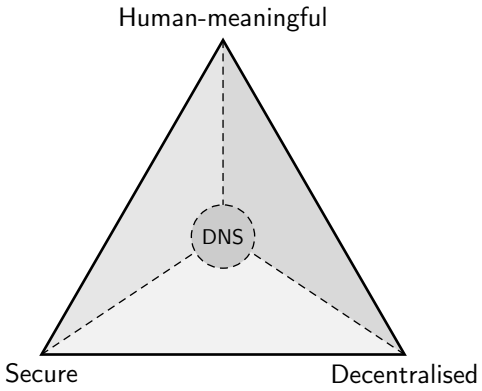


Figure 8.1: Zooko's triangle

—as occurs with the implementation of the DNSSEC extension, or the publication of the RFC 7626 entitled *DNS Privacy Considerations* [27]— further research must be performed as a result of a non-contemporary solution unable to face the new technological scenarios. With this aim in mind, several proposals have emerged as a solution to cope some of this questions, such as the decentralised solution based on the concept of blockchain and known as Namecoin [80], or the GNU Name System [148].

In this context, we also believe that it is important to remark that any future alternatives should exhibit all the three properties of Zooko's triangle (*cf.* Figure 8.1). Zooko's triangle is a diagram of three properties that are considered desirable for the names of participants in a network protocol, that is (1) *Secure*: there is one, unique and specific entity to which the name applies (2) *Decentralised:* no central authority controls all the names, and (3) *Human-meaningful*: the name is something you can actually remember instead of some long string of randomness. The triangle is part of the conjecture proposed by Zooko Wilcox-O'Hearn in 2001, and that states that out of these three properties, a naming system can only have two. This is the case of DNSSEC, which fulfils the properties of secure and human-meaningful, but it is based on a hierarchical structure with central authorities under the jurisdiction of the ICANN. In spite of this, and as Aaron Swartz argued in [138], it is possible to surpass the conjecture and construct systems that exhibit the three properties.

Taking into consideration the previous restrictions, it is clearly responsibility of the research community to determine in a broad sense the current problems, to analyse the approaches presented, and to propose a global solution capable of overcoming the aforementioned deficiencies.

8.2.2 Bonets and malicious uses of the DNS and Tor

In Chapter 3 we exposed how the lack of security of the DNS protocol is abused by the fast-flux network operators. Nevertheless, fast-flux networks is not the only strategy that can be used by the botmasters regarding the DNS protocol. One additional technique is known as Domain Generator Algorithm (DGA). DGA are algorithms included in the code executed by the bots that are capable of generating periodically new domain names. These domain names are used by bots to contact to their C&C and receive updates or new commands. Each new domain name is valid just only for certain short period of time. Since the botmaster knows the algorithm used, he can register the domains beforehand, and associate to them the C&C in some manner (*e.g.* by a simple DNS record of type A). Thus, if one of these domains is taken

down, the botmaster does not lose the control of the bots. He just only waits until the new domain name is generated by the DGA of the bots and they contact again with the C&C. This is another clear example of a DNS weakness with respect to the absence of formal verification procedures during the registration of a domain. This problem has also received attention from the research community and several results have been published [8, 69, 104, 120, 165]. Since DGA shares the same nature with the fast-flux technique, we consider this as a potential future line of research.

Beyond the techniques explained in the previous paragraph, botnets can exploit other deficiencies inherent to the DNS and DNSSEC protocols. In this case, the goal is not to protect the botnet itself, but to abuse the weaknesses of the DNS and DNSSEC protocols as a way to perpetrate DDoS attacks [118]. DNS and DNSSEC can be deployed using both TCP and UDP protocols. Since UDP protocol is susceptible of suffering IP Spoofing attacks, and that in some cases the size of the DNS responses are greater than the queries, botmasters can perform reflection attacks with a considerable amplification factors. The case is worse if we consider the DNSSEC protocol, since DNSSEC responses include the digital signatures and, therefore, the amplification can achieve factors of up to 100:1 [146]. In spite of some research contributions have been published —like the proposal of van Rijswijk-Deij *et al.* that suggest to use ECC instead of RSA [147]— it seems that nowadays the only effective mitigation mechanisms deal with distributed solutions based on Content Delivery Networks (CDN) or BGP routing [121]. We consider that further research must be conducted to tackle the problem at its root, which is the DNS protocol itself.

Aside the malicious uses of the DNS described previously, botmasters are constantly looking for new strategies that allow them to resist against the take down of their botnets. In that sense, we described in Chapter 2 several architectures with different levels of robustness. However, it is possible to go one step further and use the Tor as a C&C infrastructure, increasing the degree of robustness. For such purpose, botmasters force the bots to contact to the C&C servers by means of a hidden service within the Tor network. In spite of that this idea was presented by Dannis Brown some years ago at *DefCon18* [28], and that it seems that in the last years there have not been so much practical botnets using this strategy [31], we have recently discovered an exception. This is the case of Mirai, a botnet based on IoT devices that has received much attention, firstly because it has launched one of the largest DDoS attacks the Internet has ever seen, and secondly because it uses the Tor as a C&C infrastructure. Considering this scenario, it can be another line of future research.

8.2.3 PIR as a privacy preserving technology for DNS

In this dissertation we have used range of queries with the aim of preserving the privacy of the users when they perform DNS resolutions. This approach, as we exposed in Chapter 4, is inspired by the PIR schemes. In this field, we can classify PIR strategies in terms of their privacy guarantees and the number of servers needed for the protection they provide. From one hand, Information-Theoretic PIR schemes (ITPIR) are multi-server protocols that guarantee the privacy of the queries regardless of the computational capabilities of the servers that answer to the query. In this case, it is assumed that the database servers do not collude to determine the performed query. On the other hand, computational PIR schemes (CPIR) are based on just only one server, and where it is assumed that the server is computationally limited and unable to break hard problems. In both categories (ITPIR and CPIR), we

can find a lot of different contributions made by the research community that could be applied to the DNS resolutions. From homomorphic encryption based schemes, trapdoor permutation based protocols, to lattice based strategies, the field of PIR is wide enough to perform a deep research to analyse how these alternatives deal with the problem of DNS privacy preserving.

8.2.4 Anonymity infrastructures as a privacy preserving solution

As we have seen throughout this dissertation, the anonymity network Tor is presented as a plausible solution to preserve the privacy of users. This solution is not only relegated to the DNS resolutions, but also allows to anonymise any communication based on the TCP protocol. A key factor that can pose a problem for users is the occurrence of high latencies during the use of this anonymity network. As a matter of fact, the work presented in Chapters 5, 6 and 7 is intended to bring some improvements in this area. These contributions are focused on the selection criteria of the nodes that will form the circuit, and on the cryptographic protocol used for the circuit establishment. Despite this, other areas can be explored in order to reduce such latencies like the analysis of the influence of TCP stack [65], the remotely measurement of node loads and estimation of their capacities [110], the use of circuit-scheduling techniques [64, 79], or the traffic splitting and multiplexing through several Tor circuits [6, 156]. Thus, from a holistic view, there are several research directions that can be taken in order to address the problems of network performance, and that impacts on the user experience.

Finally, our research has been focused on the most widely used anonymity network nowadays: Tor. However, there are other alternatives [127] whose properties are interesting in comparison with Tor. Some of these alternatives —such as Freenet [44], GNUnet [19] or I2P [144]— are not merely theoretical models, and have fully functional implementations. Such empirical approaches open new opportunities not only for the theoretical study of privacy preserving technologies, but also from a pragmatic point of view.

Appendices

A

Number of walks of length λ between any two distinct vertices of a K_n graph

Let K_n be a complete graph with n vertices and $\frac{n(n-1)}{2}$ edges, such that every pair of distinct vertices is connected by a unique edge. Then, a walk in K_n of length λ from vertex v_1 to vertex $v_{\lambda+1}$ corresponds to the following sequence:

$$\underbrace{v_1 \xrightarrow{e_1} v_2 \xrightarrow{e_2} v_3 \xrightarrow{e_3} v_4 \xrightarrow{e_4} ... \xrightarrow{e_{\lambda-1}} v_\lambda \xrightarrow{e_\lambda} v_{\lambda+1}}_{\text{walk in G of length } \lambda}$$

such that each v_i is a vertex of K_n, each e_j is an edge of K_n, and the vertices connected by e_i are v_i and v_{i+1}.

Let A be the adjacency matrix of K_n, such that A is an n-square binary matrix in which each entry is either zero or one, *i.e.*, every (i, j)-entry in A is equal to the number of edges incident to v_i and v_j. Moreover, A is symmetric and circulant [16]. It has always zeros on the leading diagonal and ones off the leading diagonal. For example, the adjacency matrix of a complete graph K_4 is always equal to:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

The total number of possible walks of length λ from vertex v_i to vertex v_j is the (i, j)-entry of A^λ, *i.e.*, the matrix product, denoted by $(\cdot)$, of λ copies of A [136]. Following the above example, the number

of walks of length 2 between any two distinct vertices can be obtained directly from A^2, such that

$$A^2 = A \cdot A = \begin{bmatrix} (n-1) & (n-2) & (n-2) & (n-2) \\ (n-2) & (n-1) & (n-2) & (n-2) \\ (n-2) & (n-2) & (n-1) & (n-2) \\ (n-2) & (n-2) & (n-2) & (n-1) \end{bmatrix}$$

which leads to

$$A^2 = A \cdot A = \begin{bmatrix} 3 & 2 & 2 & 2 \\ 2 & 3 & 2 & 2 \\ 2 & 2 & 3 & 2 \\ 2 & 2 & 2 & 3 \end{bmatrix}$$

Note that any (i,j)-entry of A^2 (where $i \neq j$) gives the same number of walks of length 2 from any two distinct vertex v_i to vertex v_j. The total number of walks of length 2 between any two distinct vertices can, thus, be obtained by consecutively adding the values of every (i,j)-entry off the leading diagonal of matrix A^2 (*e.g.*, any (i,j)-entry in the upper triangle of the matrix). In the above example, it suffices to sum $\frac{4(4-1)}{2}$ times (*i.e.*, the number of edges in K_4) the value 2 that any (i,j)-entry (where $i \neq j$) has in A^2. This amounts to having exactly 12 possible walks on any K_4 graph.

Therefore, the problem of finding the number of walks of length λ between any two distinct vertices of a K_n graph reduces to finding the (i,j)-entry of A^λ, where $i \neq j$. Indeed, let $a^\lambda_{i,j}$ be the (i,j)-entry of A^λ. Then, the recurrence relation between the original adjacency matrix A, and the matrix product of up to $\lambda - 1$ copies of A, *i.e.*,

$$A^\lambda = A^{\lambda-1} \cdot A \tag{A.1}$$

with initial conditions:

$$a^2_{i,j} = \begin{cases} (n-2) & \text{if } i \neq j \\ (n-1) & \text{if } i = j \end{cases}, \; a^1_{i,j} = \begin{cases} 1 & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

is sufficient to solve the problem. Notice, moreover, that the result does not depend on any precise value of either i or j. Indeed, it is proved in [136] that there is a constant relationship between the (i,j)-entries off the leading diagonal of A^λ and the (i,j)-entries on the leading diagonal of A^λ. More precisely, let t^λ be any (i,j)-entry off the leading diagonal of A^λ (*i.e.*, $t^\lambda = a^\lambda_{i,j}$ such that $i \neq j$). Let d^λ be any (i,i)-entry on the leading diagonal of A^λ (*i.e.*, $t^\lambda = a^\lambda_{i,i}$). Then, if we subtract t^λ from d^λ, the results is always equal to $(-1)^\lambda$. In other words, if we express A^λ as follows:

$$A^\lambda = [a^\lambda_{i,j}] = \begin{cases} t^\lambda & \text{if } i \neq j \\ d^\lambda & \text{if } i = j \end{cases}$$

then $t^\lambda = d^\lambda + (-1)^\lambda$. We can now use the recurrence relation shown in Equation (A.1) to derive the following two results:

$$t^\lambda = (n-2)t^{\lambda-1} + d^{\lambda-1} \tag{A.2}$$

$$d^\lambda = (n-1)t^{\lambda-1} \tag{A.3}$$

with the initial conditions $t^1 = 1$ and $d^1 = 0$.

Cumbersome, but elementary, transformations shown in both [16] and [136] lead us to unfold the two recurrence relations in both Equation (A.2) and (A.3) to the following two self-contained expressions:

$$t^\lambda = \frac{(n-1)^\lambda - (-1)^\lambda}{n} \tag{A.4}$$

$$d^\lambda = \frac{(n-1)^\lambda + (n-1)(-1)^\lambda}{n} \tag{A.5}$$

To conclude, we can now use Equations (A.4) and (A.5) to express the total number of closed and non-closed walks in the complete graph K_n by simply adding to them the number of edges in the graph (*i.e.*, $\frac{n(n-1)}{2}$). From Equation (A.4) we have now the value of any (i,j)-entry in A^λ such that $i \neq j$. As we did previously in the example of the complete graph K_4, the total number of walks of length λ between any two distinct vertices can be obtained by consecutively adding $\frac{n(n-1)}{2}$ times the values of any of the (i,j)-entries off the leading diagonal of matrix A^λ. This amounts to having exactly $\frac{n(n-1)}{2} \cdot t^\lambda$ which simplifying leads to:

$$\frac{(n-1)((n-1)^\lambda - (-1)^\lambda)}{2} \tag{A.6}$$

possible walks of length λ on any K_n graph.

Bibliography

[1] T. ABEEL, Y. VAN DE PEER, and Y. SAEYS. Java-ML: A Machine Learning Library. In: *ournal of Machine Learning Research*, **10**: (June 2009), 931–934 (see p. 31)

[2] B. AGER, H. DREGER, and A. FELDMANN. "Predicting the DNSSEC overhead using DNS traces". In: *2006 40th Annual Conference on Information Sciences and Systems*. Institute of Electrical and Electronics Engineers (IEEE), Mar. 2006, 1484–1489. DOI: 10.1109/CISS.2006.286699 (see p. 44)

[3] H. AL-ASSAM and S. JASSIM. Security evaluation of biometric keys. In: *Computers & Security*, **31**:2 (Mar. 2012), 151–163. DOI: 10.1016/j.cose.2012.01.002 (see p. 54)

[4] *Alexa Top 500 Global Sites*. URL: http://www.alexa.com/topsites (see p. 31)

[5] M. ALSABAH and I. GOLDBERG. Performance and Security Improvements for Tor: A Survey. In: *ACM Computing Surveys*, **49**:2 (Sept. 2016), 1–36. DOI: 10.1145/2946802 (see p. 3)

[6] M. ALSABAH, K. BAUER, T. ELAHI, and I. GOLDBERG. "The Path Less Travelled: Overcoming Tor's Bottlenecks with Traffic Splitting". In: *Proceedings of the 13th Privacy Enhancing Technologies Symposium (PETS 2013)*. July 2013 (see p. 106)

[7] M. ANDERBERG. *Cluster Analysis for Applications*. Tech. rep. DTIC Document, Academic Press, 1973 (see p. 73)

[8] M. ANTONAKAKIS, R. PERDISCI, D. DAGON, W. LEE, and N. FEAMSTER. "Building a Dynamic Reputation System for DNS". In: *Proceedings of the 19th USENIX conference on Security*. USENIX Security'10. Washington, DC: USENIX Association, 2010, 18–18 (see p. 105)

[9] *Arbor Summary Report on Global Fast Flux*. URL: http://atlas.arbor.net/summary/fastflux (see p. 31)

[10] M. ARLITT and C. WILLIAMSON. An analysis of TCP reset behaviour on the internet. In: *ACM SIGCOMM Computer Communication Review*, **35**:1 (Jan. 2005), 37. DOI: 10.1145/1052812.1052823 (see p. 41)

[11] A. ARNELLOS, D. LEKKAS, D. ZISSIS, T. SPYROU, and J. DARZENTAS. Fair digital signing: The structural reliability of signed documents. In: *Computers & Security*, **30**:8 (Nov. 2011), 580–596. DOI: 10.1016/j.cose.2011.09.001 (see p. 54)

[12] D. ATKINS and R. AUSTEIN. *Threat Analysis of the Domain Name System (DNS)*. Tech. rep. 3833. Aug. 2004. 16 pp. DOI: 10.17487/rfc3833 (see pp. 2, 37)

[13] A. BACK, U. MÖLLER, and A. STIGLIC. "Traffic Analysis Attacks and Trade-Offs in Anonymity Providing Systems". In: *Proceedings of the 4th International Workshop on Information Hiding*. IHW'01. London, UK, UK: Springer-Verlag, 2001, 245–257. DOI: 10.1007/3-540-45496-9_18 (see p. 52)

[14] M. BACKES, A. KATE, and E. MOHAMMADI. "Ace: An Efficient Key-exchange Protocol for Onion Routing". In: *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*. WPES '12. New York, NY, USA: Association for Computing Machinery (ACM), 2012, 55–64. DOI: 10.1145/2381966.2381974 (see p. 86)

[15] E. BARKER. Recommendation for Key Management Part 1: General. In: (Jan. 2016). DOI: 10.6028/nist.sp.800-57pt1r4 (see p. 98)

[16] P. BARRY. On Integer Sequences Associated with the Cyclic and Complete Graphs. In: *Journal of Integer Sequences*, **10**: (4 2007). Article 07.4.8 (see pp. 109, 111)

[17] K. BAUER, D. MCCOY, D. GRUNWALD, T. KOHNO, and D. SICKER. "Low-Resource Routing Attacks Against Tor". In: *2007 ACM workshop on Privacy in electronic society (WPES'07)*. New York, NY, USA: Association for Computing Machinery (ACM), 2007, 11–20. DOI: 10.1145/1314333.1314336 (see pp. 41, 46, 47, 81, 82)

[18] M. BELLARE and P. ROGAWAY. "Provably Secure Session Key Distribution: The Three Party Case". In: *Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing*. STOC '95. New York, NY, USA: ACM, 1995, 57–66. DOI: 10.1145/225058.225084 (see p. 92)

[19] K. BENNETT, T. STEF, C. GROTHOFF, T. HOROZOV, and I. PATRASCU. *The GNet Whitepaper*. Tech. rep. Purdue University, 2002 (see p. 106)

[20] J. BETHENCOURT, A. SAHAI, and B. WATERS. "Ciphertext-Policy Attribute-Based Encryption". In: *Proceedings of the 2007 IEEE Symposium on Security and Privacy*. SP '07. Washington, DC, USA: IEEE Computer Society, 2007, 321–334. DOI: 10.1109/SP.2007.11 (see p. 88)

[21] A. BIRYUKOV and I. PUSTOGAROV. "Bitcoin over Tor isn't a Good Idea". In: *2015 IEEE Symposium on Security and Privacy*. Institute of Electrical and Electronics Engineers (IEEE), May 2015, 122–134. DOI: 10.1109/SP.2015.15 (see p. 1)

[22] J.-Y. BISIAUX. Feature: DNS Threats and Mitigation Strategies. In: *Network Security*, **2014**:7 (July 2014), 5–9. DOI: 10.1016/S1353-4858(14)70068-6 (see p. 2)

[23] D. BONEH, X. BOYEN, and E.-J. GOH. *Hierarchical Identity Based Encryption with Constant Size Ciphertext*. In: *Advances in Cryptology – EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, 440–456. DOI: 10.1007/11426639_26 (see p. 89)

[24] D. BONEH and M. FRANKLIN. "Identity-Based Encryption from the Weil Pairing". In: *Advances in Cryptology — CRYPTO 2001: 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19–23, 2001 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, 213–229. DOI: 10.1007/3-540-44647-8_13 (see pp. 13, 86)

[25] D. BONEH, B. LYNN, and H. SHACHAM. Short Signatures from the Weil Pairing. English. In: *Journal of Cryptology*, **17**:4 (July 2004), 297–319. DOI: 10.1007/s00145-004-0314-9 (see p. 90)

[26] N. BORISOV, G. DANEZIS, P. MITTAL, and P. TABRIZ. "Denial of Service or Denial of Security?" In: *Proceedings of the 14th ACM Conference on Computer and Communications Security*. CCS '07. New York, NY, USA: Association for Computing Machinery (ACM), 2007, 92–102. DOI: 10.1145/1315245.1315258 (see p. 45)

[27] S. BORTZMEYER. *DNS Privacy Considerations*. Tech. rep. 7626. Aug. 2015. 17 pp. DOI: 10.17487/rfc7626 (see p. 104)

[28] D. BROWN. Resilient botnet command and control with Tor. In: *DEF CON*, **18**: (2010) (see p. 105)

[29] J. CAMENISCH and A. LYSYANSKAYA. "A Formal Treatment of Onion Routing". In: *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*. Vol. 3621. Lecture Notes in Computer Science. Santa Barbara, California: Springer, 2005, 169–187. DOI: 10.1007/11535218_11 (see pp. 94, 95)

[30] R. CANETTI, S. HALEVI, and J. KATZ. *A Forward-Secure Public-Key Encryption Scheme*. In: *Advances in Cryptology — EUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, 255–271. DOI: 10.1007/3-540-39200-9_16 (see p. 89)

[31] M. CASENOVE and A. MIRAGLIA. Botnet over Tor: The illusion of hiding. In: *2014 6th International Conference On Cyber Conflict (CyCon 2014)*, (June 2014), 273–282. DOI: 10.1109/cycon.2014.6916408 (see p. 105)

[32] S. CASTILLO-PÉREZ and J. GARCIA-ALFARO. On the Use of Latency Graphs for the Construction of Tor Circuits. In: *CoRR*, **abs/1208.3730**: (Aug. 2012) (see p. 4)

[33] S. CASTILLO-PÉREZ and J. GARCIA-ALFARO. Onion Routing Circuit Construction via Latency Graphs. In: *Computers & Security*, **37**: (Sept. 2013), 197–214. DOI: 10.1016/j.cose.2013.03.003 (see pp. 4, 96)

[34] S. CASTILLO-PÉREZ, J. GARCIA-ALFARO, and J. BORRELL-VIADER. A Scalable and Single-Pass Authenticated Key Agreement Protocol for the Establishment of Second-Generation Onion Routing Circuits. In: *To be submitted*, (2017) (see p. 4)

[35] S. CASTILLO-PÉREZ and J. GARCIA-ALFARO. *Anonymous Resolution of DNS Queries*. In: *On the Move to Meaningful Internet Systems: OTM 2008*. Ed. by R. MEERSMAN and Z. TARI. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, 987–1000. DOI: 10.1007/978-3-540-88873-4_5 (see p. 4)

[36] S. CASTILLO-PÉREZ and J. GARCIA-ALFARO. Evaluation of Two Privacy-Preserving Protocols for the DNS. In: *2009 6th International Conference on Information Technology: New Generations*, (2009), 411–416. DOI: 10.1109/ITNG.2009.195 (see p. 4)

[37] D. CATALANO, D. FIORE, and R. GENNARO. "Certificateless Onion Routing". In: *Proceedings of the 16th ACM Conference on Computer and Communications Security*. CCS '09. New York, NY, USA: Association for Computing Machinery (ACM), 2009, 151–160. DOI: 10.1145/1653662.1653682 (see pp. 87, 94, 98)

[38] D. CATALANO, D. FIORE, and R. GENNARO. A certificateless approach to onion routing. In: *International Journal of Information Security*, (June 2016), 1–17. DOI: 10.1007/s10207-016-0337-x (see pp. 87, 98)

[39] D. CATALANO, M. DI RAIMONDO, D. FIORE, R. GENNARO, and O. PUGLISI. Fully non-interactive onion routing with forward secrecy. English. In: *International Journal of Information Security*, **12**:1 (Dec. 2012), 33–47. DOI: 10.1007/s10207-012-0185-2 (see p. 88)

[40] D. L. CHAUM. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. In: *Communications of the ACM*, **24**:2 (Feb. 1981), 84–90. DOI: 10.1145/358549.358563 (see p. 18)

[41] L. CHEN and C. KUDLA. "Identity Based Authenticated Key Agreement Protocols from Pairings". In: *16th IEEE Computer Security Foundations Workshop, 2003. Proceedings.* IEEE Computer Society Press, 2002, 219–233. DOI: 10.1109/CSFW.2003.1212715 (see pp. 89, 92, 95)

[42] B. CHOR, E. KUSHILEVITZ, O. GOLDREICH, and M. SUDAN. Private Information Retrieval. In: *Journal of the ACM*, **45**:6 (Nov. 1998), 965–981. DOI: 10.1145/293347.293350 (see pp. 2, 41)

[43] B. CHUN, D. CULLER, T. ROSCOE, A. BAVIER, L. PETERSON, M. WAWRZONIAK, and M. BOWMAN. PlanetLab: An Overlay Testbed for Broad-Coverage Services. In: *ACM SIGCOMM Computer Communication Review*, **33**:3 (July 2003), 3. DOI: 10.1145/956993.956995 (see p. 73)

[44] I. CLARKE, O. SANDBERG, B. WILEY, and T. W. HONG. "Freenet: A Distributed Anonymous Information Storage and Retrieval System". In: *International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*. New York, NY, USA: Springer-Verlag New York, Inc., 2001, 46–66 (see pp. 1, 106)

[45] S. CLAUSS and S. SCHIFFNER. "Structuring Anonymity Metrics". In: *Proceedings of the second ACM workshop on Digital identity management*. DIM '06. ACM. New York, NY, USA: Association for Computing Machinery (ACM), 2006, 55–62. DOI: 10.1145/1179529.1179539 (see p. 81)

[46] M. COATES, A. HERO, R. NOWAK, and B. YU. Internet Tomography. In: *IEEE Signal Processing Magazine*, **19**:3 (May 2002), 47–65. DOI: 10.1109/79.998081 (see p. 64)

[47] F. T. COMMISSION et al. "Protecting consumers from spam, spyware, and fraud". In: *A Legislative Recommendation to Congress*. 2005 (see p. 38)

[48] E. COOKE, F. JAHANIAN, and D. MCPHERSON. "The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets". In: *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop*. SRUTI'05. Cambridge, MA: USENIX Association, 2005, 6–6 (see p. 18)

[49] G. DANEZIS and I. GOLDBERG. "Sphinx: A Compact and Provably Secure Mix Format". In: *2009 30th IEEE Symposium on Security and Privacy*. Institute of Electrical and Electronics Engineers (IEEE), May 2009, 269–282. DOI: 10.1109/SP.2009.15 (see p. 95)

[50] C. DÍAZ, S. SEYS, J. CLAESSENS, and B. PRENEEL. "Towards Measuring Anonymity". In: *Proceedings of the 2nd international conference on Privacy enhancing technologies*. PET'02. Berlin, Heidelberg: Springer-Verlag, 2003, 54–68. DOI: 10.1007/3-540-36467-6_5 (see pp. 46, 53, 81)

[51] R. DINGLEDINE and N. MATHEWSON. *Tor protocol specification*. 2003-2017. URL: https://gitweb.torproject.org/torspec.git/tree/tor-spec.txt (see p. 98)

[52] R. DINGLEDINE, N. MATHEWSON, and P. SYVERSON. "Tor: The Second-generation Onion Router". In: *Proceedings of the 13th conference on USENIX Security Symposium - Volume 13*. SSYM'04. Berkeley, CA, USA: USENIX Association, 2004, 21–21 (see pp. 1, 2, 20, 38, 47, 85, 98)

[53] R. DINGLEDINE, M. PERRY, N. MATHEWSON, D. JOHNSON, D. FIFIELD, G. KADIANAKIS, J. APPELBAUM, K. LOESING, L. NORDBERG, R. RANSOM, and S. HAHN. *Tor directory protocol, version 3*. 2005-2017. URL: https://gitweb.torproject.org/torspec.git/tree/dir-spec.txt (see p. 73)

[54] N. DORASWAMY, K. R. GLENN, and R. L. THAYER. *IP Security Document Roadmap*. Tech. rep. 2411. Nov. 1998. 11 pp. DOI: 10.17487/rfc2411 (see p. 39)

[55] N. DOSHI and D. JINWALA. "AB-OR: Improving the Efficiency in Onion Routing Using Attribute Based Cryptography". English. In: *Computer Networks & Communications (NetCom)*. Ed. by N. CHAKI, N. MEGHANATHAN, and D. NAGAMALAI. Vol. 131. Lecture Notes in Electrical Engineering. Springer New York, 2013, 425–432. DOI: 10.1007/978-1-4614-6154-8_42 (see p. 88)

[56] R. DUPONT and A. ENGE. Provably Secure Non-interactive Key Distribution Based on Pairings. In: *Discrete Applied Mathematics*, **154**:2 (Feb. 2006), 270–276. DOI: 10.1016/j.dam.2005.03.024 (see p. 87)

[57] ECRYPT. Yearly Report on Algorithms and Keysizes (2011-2012), Rev. 1.0. In: *ECRYPT II Network of Excellence (NoE), funded within the Information Societies Technology (IST) Programme of the European Commission's Seventh Framework Programme (FP7)*, (2012) (see p. 98)

[58] M. EDMAN, F. SIVRIKAYA, and B. YENER. "A Combinatorial Approach to Measuring Anonymity". In: *2007 IEEE Intelligence and Security Informatics*. IEEE. Institute of Electrical and Electronics Engineers (IEEE), May 2007, 356–363. DOI: 10.1109/ISI.2007.379497 (see p. 81)

[59] M. EDMAN and P. SYVERSON. "As-awareness in Tor Path Selection". In: *Proceedings of the 16th ACM conference on Computer and communications security*. CCS'09. New York, NY, USA: Association for Computing Machinery (ACM), 2009, 380–389. DOI: 10.1145/1653662.1653708 (see p. 81)

[60] M. FEILY, A. SHAHRESTANI, and S. RAMADASS. "A Survey of Botnet and Botnet Detection". In: *2009 Third International Conference on Emerging Security Information, Systems and Technologies*. Institute of Electrical and Electronics Engineers (IEEE), June 2009, 268–273. DOI: 10.1109/SECURWARE.2009.48 (see p. 17)

[61] L. C. FREEMAN. A Set of Measures of Centrality Based on Betweenness. In: *Sociometry*, **40**:1 (Mar. 1977), 35–41 (see p. 69)

[62] J. GARCIA-ALFARO, M. BARBEAU, and E. KRANAKIS. "Evaluation of Anonymized ONS Queries". In: *Workshop on Security of Autonomous and Spontaneous Networks (SETOP 2008)*. 2008, 47–60 (see pp. 38, 82)

[63] J. GARCIA-ALFARO and S. CASTILLO-PÉREZ. "Resolution of anonymous DNS queries (In Spanish)". In: *X Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*. Sept. 2008 (see p. 4)

[64] K. T. GIRRY, S. OHZAHATA, C. WU, and T. KATO. "A Circuit Switching Method for Improving Congestion of Tor Network". In: *2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications*. Nov. 2014, 416–421. DOI: 10.1109/BWCCA.2014.97 (see p. 106)

[65] K. T. GIRRY, S. OHZAHATA, C. WU, and T. KATO. "Analyzing the drawbacks of node-based delays in Tor". In: *2014 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*. Institute of Electrical and Electronics Engineers (IEEE), May 2014, 1–6. DOI: 10.1109/CQR.2014.7152451 (see p. 106)

[66] I. GOLDBERG. "On the Security of the Tor Authentication Protocol". In: *Proceedings of the 6th International Conference on Privacy Enhancing Technologies*. PET'06. Berlin, Heidelberg: Springer-Verlag, 2006, 316–331. DOI: 10.1007/11957454_18 (see pp. 20, 85)

[67] I. GOLDBERG, D. STEBILA, and B. USTAOGLU. Anonymity and one-way authentication in key exchange protocols. In: *Designs, Codes and Cryptography*, **67**:2 (Jan. 2013), 245–269. DOI: 10.1007/s10623-011-9604-z (see p. 86)

[68] D. M. GOLDSCHLAG, M. G. REED, and P. F. SYVERSON. "Hiding Routing information". In: *Information Hiding: First International Workshop Cambridge, U.K., May 30 – June 1, 1996 Proceedings*. Ed. by R. ANDERSON. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, 137–150. DOI: 10.1007/3-540-61996-8_37 (see p. 19)

[69] M. GRILL, I. NIKOLAEV, V. VALEROS, and M. REHAK. "Detecting DGA malware using NetFlow". In: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. Institute of Electrical and Electronics Engineers (IEEE), May 2015, 1304–1309. DOI: 10.1109/INM.2015.7140486 (see p. 105)

[70] A. A. HAGBERG, D. A. SCHULT, and P. J. SWART. "Exploring Network Structure, Dynamics, and Function Using NetworkX". In: *Proceedings of the 7th Python in Science Conference (SciPy2008)*. Pasadena, CA USA, Aug. 2008, 11–15 (see p. 75)

[71] A. HAMEL, J. GRÉGOIRE, and I. GOLDBERG. *The Mis-entropists: New Approaches to Measures in Tor*. Tech. rep. 18. University of Waterloo, 2011 (see p. 81)

[72] B. HOENEISEN, A. MAYRHOFER, and J. LIVINGOOD. *IANA Registration of Enumservices: Guide, Template, and IANA Considerations*. Tech. rep. 6117. Mar. 2011. 40 pp. DOI: 10.17487/rfc6117 (see pp. 2, 36)

[73] T. HOLZ, C. GORECKI, K. RIECK, and F. C. FREILING. "Measuring and Detecting Fast-Flux Service Networks". In: *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. 2008 (see pp. 3, 25, 26)

[74] N. HOPPER, E. Y. VASSERMAN, and E. CHAN-TIN. How Much Anonymity does Network Latency Leak? In: *ACM Transactions on Information and System Security (TISSEC)*, **13**:2 (Mar. 2010), 1–28. DOI: 10.1145/1698750.1698753 (see p. 52)

[75] C.-H. HSU, C.-Y. HUANG, and K.-T. CHEN. "Fast-Flux Bot Detection in Real Time". In: *Recent Advances in Intrusion Detection*. Ed. by S. JHA, R. SOMMER, and C. KREIBICH. Vol. 6307. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, 464–483. DOI: 10.1007/978-3-642-15512-3_24 (see pp. 26, 27, 31, 32, 102)

[76] J. HUNTER. The Exponentially Weighted Moving Average. In: *Journal of Quality Technology*, **18**:4 (1986), 203–210 (see p. 65)

[77] A. JOHNSTON, V. VENKATARAMANAN, and M. SOROUSHNEJAD. *Shared Appearances of a Session Initiation Protocol (SIP) Address of Record (AOR)*. Tech. rep. 7463. Mar. 2015. 72 pp. DOI: 10.17487/rfc7463 (see p. 37)

[78] A. JOUX. A One Round Protocol for Tripartite Diffie–Hellman. In: *Journal of Cryptology*, **17**:4 (June 2004), 263–276. DOI: 10.1007/s00145-004-0312-y (see p. 13)

[79] T. G. KALE, S. OHZAHATA, C. WU, and T. KATO. "Evaluating tor modified switching algorithm in the emulation environment". In: *2016 22nd Asia-Pacific Conference on Communications (APCC)*. Institute of Electrical and Electronics Engineers (IEEE), Aug. 2016, 510–516. DOI: 10.1109/APCC.2016.7581478 (see p. 106)

[80] H. KALODNER, M. CARLSTEN, P. ELLENBOGEN, J. BONNEAU, and A. NARAYANAN. "An empirical study of Namecoin and lessons for decentralized namespace design". In: *Workshop on the Economics of Information Security (WEIS)*. 2015 (see p. 104)

[81] A. KATE and I. GOLDBERG. *Using Sphinx to Improve Onion Routing Circuit Construction*. In: *Financial Cryptography and Data Security: 14th International Conference, FC 2010, Tenerife, Canary Islands, January 25-28, 2010, Revised Selected Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, 359–366. DOI: 10.1007/978-3-642-14577-3_30 (see p. 95)

[82] A. KATE, G. ZAVERUCHA, and I. GOLDBERG. "Pairing-Based Onion Routing". English. In: *Privacy Enhancing Technologies*. Ed. by N. BORISOV and P. GOLLE. Vol. 4776. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, 95–112. DOI: 10.1007/978-3-540-75551-7_7 (see pp. 86, 88, 98)

[83] A. KATE, G. M. ZAVERUCHA, and I. GOLDBERG. Pairing-Based Onion Routing with Improved Forward Secrecy. In: *ACM Transactions on Information and System Security*, **13**:4 (Dec. 2010), 1–32. DOI: 10.1145/1880022.1880023 (see pp. 86, 89, 91, 94, 95, 97, 98, 103)

[84] H. KIM and J. H. HUH. PIN selection policies: Are they really effective? In: *Computers & Security*, **31**:4 (June 2012), 484–496. DOI: 10.1016/j.cose.2012.02.003 (see p. 54)

[85] M. KNYSZ, X. HU, and K. G. SHIN. "Good guys vs. Bot Guise: Mimicry attacks against fast-flux detection systems". In: *2011 Proceedings IEEE INFOCOM*. Institute of Electrical & Electronics Engineers (IEEE), Apr. 2011, 1844–1852. DOI: 10.1109/INFCOM.2011.5934985 (see p. 26)

[86] R. KOHAVI. "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection". In: *International Joint Conference on Artificial Intelligence*. 1995, 1137–1145 (see p. 31)

[87] M. KONTE, N. FEAMSTER, and J. JUNG. "Dynamics of Online Scam Hosting Infrastructure". In: *Passive and Active Network Measurement*. Ed. by S. MOON, R. TEIXEIRA, and S. UHLIG. Vol. 5448. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, 219–228. DOI: 10.1007/978-3-642-00975-4_22 (see p. 25)

[88] S. KÖPSELL. "Low Latency Anonymous Communication – How Long Are Users Willing to Wait?" In: *Emerging Trends in Information and Communication Security*. Ed. by G. MÜLLER. Vol. 3995. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2006, 221–237. DOI: 10.1007/11766155_16 (see p. 79)

[89] B. LAMACCHIA, K. LAUTER, and A. MITYAGIN. "Stronger Security of Authenticated Key Exchange". In: *Proceedings of the 1st International Conference on Provable Security*. ProvSec'07. Wollongong, Australia: Springer-Verlag, 2007, 1–16 (see p. 86)

[90] L. LAW, A. MENEZES, M. QU, J. SOLINAS, and S. VANSTONE. An Efficient Protocol for Authenticated Key Agreement. In: *Designs, Codes and Cryptography*, **28**:2 (Mar. 2003), 119–134. DOI: 10.1023/A:1022595222606 (see p. 84)

[91] B. LYNN. *PBC - The Pairing-Based Cryptography Library*. URL: https://crypto.stanford.edu/pbc/ (see p. 98)

[92] J. MACQUEEN. "Some Methods for Classification and Analysis of Multivariate Observations". In: *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability*. Vol. 1. California, USA. University of California Press, 1967, 281–297 (see p. 73)

[93] S. MARTINEZ-BEA, S. CASTILLO-PEREZ, and J. GARCIA-ALFARO. "Real-time malicious fast-flux detection using DNS and bot related features". In: *2013 Eleventh Annual Conference on Privacy, Security and Trust*. Institute of Electrical and Electronics Engineers (IEEE), July 2013, 369–372. DOI: 10.1109/PST.2013.6596093 (see p. 4)

[94] N. MATHEWSON, R. DINGLEDINE, M. PERRY, D. JOHNSON, H. BOCK, and T. TOUCEDA. *Python Tor controller*. 2005-2011. URL: https://gitweb.torproject.org/atagar/pytorctl.git (see p. 75)

[95] D. K. MCGRATH, A. KALAFUT, and M. GUPTA. Phishing Infrastructure Fluxes All the Way. In: *IEEE Security & Privacy Magazine*, **7**:5 (Sept. 2009), 21–28. DOI: 10.1109/MSP.2009.130 (see pp. 26, 27, 31, 32, 102)

[96] M. H. MEALLING and D. R. DANIEL. *The Naming Authority Pointer (NAPTR) DNS Resource Record*. Tech. rep. 2915. Sept. 2000. 18 pp. DOI: 10.17487/rfc2915 (see pp. 36, 37)

[97] S. P. MEENAKSHI and S. V. RAGHAVAN. "Impact of IPSec Overhead on Web Application Servers". In: *2006 International Conference on Advanced Computing and Communications*. Institute of Electrical and Electronics Engineers (IEEE), Dec. 2006, 652–657. DOI: 10.1109/ADCOM.2006.4289981 (see p. 39)

[98] P. MOCKAPETRIS. *Domain names - implementation and specification*. Tech. rep. 1035. Nov. 1987. 55 pp. DOI: 10.17487/rfc1035 (see p. 14)

[99] S. J. MURDOCH. "Hot or Not: Revealing Hidden Services by Their Clock Skew". In: *Proceedings of the 13th ACM conference on Computer and communications security*. CCS'06. New York, NY, USA: Association for Computing Machinery (ACM), 2006, 27–36. DOI: 10.1145/1180405.1180410 (see pp. 40, 52)

[100] S. J. MURDOCH and G. DANEZIS. "Low-Cost Traffic Analysis of Tor". In: *Proceedings of the 2005 IEEE Symposium on Security and Privacy*. SP'05. Washington, DC, USA: IEEE Computer Society, 2005, 183–195. DOI: 10.1109/SP.2005.12 (see p. 40)

[101] S. J. MURDOCH and R. N. WATSON. "Metrics for Security and Performance in Low-Latency Anonymity Systems". In: *Proceedings of the 8th International Symposium on Privacy Enhancing Technologies*. PETS'08. Berlin, Heidelberg: Springer-Verlag, 2008, 115–132. DOI: 10.1007/978-3-540-70630-4_8 (see p. 81)

[102] M. F. M. MURSI, G. M. R. ASSASSA, A. ABDELHAFEZ, and K. M. A. SAMRA. On the Development of Electronic Voting: A Survey. In: *International Journal of Computer Applications*, **61**:16 (Jan. 2013), 1–11 (see p. 1)

[103] S. NAKAMOTO. *Bitcoin: A peer-to-peer electronic cash system*. 2008 (see p. 1)

[104] T.-D. NGUYEN, T.-D. CAO, and L.-G. NGUYEN. "DGA Botnet Detection Using Collaborative Filtering and Density-based Clustering". In: *Proceedings of the Sixth International Symposium on Information and Communication Technology*. SoICT 2015. New York, NY, USA: Association for Computing Machinery (ACM), 2015, 203–209. DOI: 10.1145/2833258.2833310 (see p. 105)

[105] NOMIUM INC. *A DNS toolkit for Python*. URL: http://www.dnspython.org/ (see p. 47)

[106] T. OKAMOTO, R. TSO, and E. OKAMOTO. "One-Way and Two-Party Authenticated ID-Based Key Agreement Protocols Using Pairing". English. In: *Modeling Decisions for Artificial Intelligence*. Vol. 3558. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, 122–133. DOI: 10.1007/11526018_13 (see p. 84)

[107] R. OSTROVSKY and W. E. SKEITH III. "A Survey of Single-Database Private Information Retrieval: Techniques and Applications". In: *Proceedings of the 10th International Conference on Practice and Theory in Public-key Cryptography*. PKC'07. Berlin, Heidelberg: Springer-Verlag, 2007, 393–411. DOI: 10.1007/978-3-540-71677-8_26 (see pp. 38, 41)

[108] L. ØVERLIER and P. SYVERSON. "Locating Hidden Servers". In: *Proceedings of the 2006 IEEE Symposium on Security and Privacy*. IEEE. Washington, DC, USA: IEEE Computer Society, 2006, 100–114. DOI: 10.1109/SP.2006.24 (see p. 40)

[109] L. ØVERLIER and P. SYVERSON. "Improving Efficiency and Simplicity of Tor Circuit Establishment and Hidden Services". In: *Proceedings of the 7th International Conference on Privacy Enhancing Technologies*. PET'07. Berlin, Heidelberg: Springer-Verlag, 2007, 134–152. DOI: 10.1007/978-3-540-75551-7_9 (see p. 85)

[110] A. PANCHENKO, F. LANZE, and T. ENGEL. "Improving Performance and Anonymity in the Tor Network". In: *Proceedings of the 31st IEEE International Performance Computing and Communications Conference (IPCCC 2012)*. Dec. 2012 (see p. 106)

[111] A. PANCHENKO and J. RENNER. "Path Selection Metrics for Performance-Improved Onion Routing". In: *9th Annual International Symposium on Applications and the Internet (SAINT'09)*. IEEE. Institute of Electrical & Electronics Engineers (IEEE), July 2009, 114–120. DOI: 10.1109/SAINT.2009.26 (see p. 82)

[112] K. PENG. "Efficiency Optimisation of Tor Using Diffie-Hellman Chain". In: *Proceedings of ICN 2011, The Tenth International Conference on Networks*. St. Maarten, The Netherlands Antilles, Jan. 2011, 41–46 (see pp. 87, 98)

[113] L. PETERSON, S. MUIR, T. ROSCOE, and A. KLINGAMAN. *PlanetLab Architecture: An Overview*. Tech. rep. PDN–06–031. PlanetLab Consortium, May 2006 (see pp. 47, 73)

[114] S. RAMACHANDRAN. *Web metrics: Size and number of resources*. May 2010. URL: https://developers.google.com/speed/articles/web-metrics (see p. 77)

[115] M. G. REED, P. F. SYVERSON, and D. M. GOLDSCHLAG. Anonymous connections and onion routing. In: *IEEE Journal on Selected Areas in Communications*, **16**:4 (May 1998), 482–494. DOI: 10.1109/49.668972 (see p. 38)

[116] J. E. Y. ROSSEBØ, S. CADZOW, and P. SIJBEN. "eTVRA, a Threat, Vulnerability and Risk Assessment Tool for eEurope". In: *Trust Management*. Ed. by K. STØLEN, W. WINSBOROUGH, F. MARTINELLI, and F. MASSACCI. Vol. 3986. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2006, 467–471. DOI: 10.1007/11755593_38 (see pp. 2, 14, 37)

[117] J. E. Y. ROSSEBØ, S. CADZOW, and P. SIJBEN. "eTVRA, a Threat, Vulnerability and Risk Assessment Method and Tool for eEurope". In: *Proceedings of the The Second International Conference on Availability, Reliability and Security*. ARES '07. Washington, DC, USA: IEEE Computer Society, 2007, 925–933. DOI: 10.1109/ARES.2007.82 (see pp. 2, 14, 37)

[118] C. ROSSOW. "Amplification Hell: Revisiting Network Protocols for DDoS Abuse". In: *Proceedings of the 2014 Network and Distributed System Security (NDSS) Symposium*. Feb. 2014 (see p. 105)

[119] R. SAKAI. "Cryptosystems based on pairings". In: *Symposium on Cryptography and Information Security 2000, SCIS2000*. Okinawa, Japan, Jan. 2000, 26–28 (see p. 86)

[120] S. SCHIAVONI, F. MAGGI, L. CAVALLARO, and S. ZANERO. *Phoenix: DGA-Based Botnet Tracking and Intelligence*. In: *Detection of Intrusions and Malware, and Vulnerability Assessment: 11th International Conference, DIMVA 2014, Egham, UK, July 10-11, 2014. Proceedings*. Ed. by S. DIETRICH. Cham: Springer International Publishing, 2014, 192–211. DOI: 10.1007/978-3-319-08509-8_11 (see p. 105)

[121] C. SCHUTIJSER. Comparing DDoS Mitigation Techniques. In: *24th Twente Student Conference on IT*, (Jan. 2016) (see p. 105)

[122] A. SERJANTOV and G. DANEZIS. *Towards an Information Theoretic Metric for Anonymity*. In: *Proceedings of the 2nd international conference on Privacy enhancing technologies*. PET'02. Berlin, Heidelberg: Springer-Verlag, 2003, 41–53. DOI: 10.1007/3-540-36467-6_4 (see pp. 46, 53, 81)

[123] A. SHAMIR. How to Share a Secret. In: *Communications of the ACM*, **22**:11 (Nov. 1979), 612–613. DOI: 10.1145/359168.359176 (see p. 87)

[124] A. SHAMIR. "Identity-Based Cryptosystems and Signature Schemes". In: *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*. Vol. 196. Lecture Notes in Computer Science. Springer, 1984, 47–53. DOI: 10.1007/3-540-39568-7_5 (see p. 97)

[125] C. E. SHANNON. A Mathematical Theory of Communication. In: *Bell system technical journal*, **27**: (1948), 379–423 (see pp. 46, 53)

[126] M. SHERR, M. BLAZE, and B. T. LOO. "Scalable Link-Based Relay Selection for Anonymous Routing". In: *Proceedings of the 9th International Symposium on Privacy Enhancing Technologies*. PETS'09. Berlin, Heidelberg: Springer-Verlag, 2009, 73–93. DOI: 10.1007/978-3-642-03168-7_5 (see p. 82)

[127] F. SHIRAZI, M. SIMEONOVSKI, M. R. ASGHAR, M. BACKES, and C. DIAZ. A Survey on Routing in Anonymous Communication Protocols. In: *arXiv preprint arXiv:1608.05538*, (2016) (see p. 106)

[128] V. SHMATIKOV and M.-H. WANG. "Measuring Relationship Anonymity in Mix Networks". In: *Proceedings of the 5th ACM workshop on Privacy in electronic society*. WPES'06. New York, NY, USA: Association for Computing Machinery (ACM), 2006, 59–62. DOI: 10.1145/1179601.1179611 (see p. 81)

[129] V. SHOUP. *On Formal Models for Secure Key Exchange*. Cryptology ePrint Archive, Report 1999/012. 1999 (see p. 85)

[130] S. S. C. SILVA, R. M. P. SILVA, R. C. G. PINTO, and R. M. SALLES. Botnets: A Survey. In: *Computers Netwowrks*, **57**:2 (Feb. 2013), 378–403. DOI: 10.1016/j.comnet.2012.07.021 (see p. 16)

[131] A. SINGH, O. NORDSTRÖM, C. LU, and A. L.M. D. SANTOS. "Malicious ICMP Tunneling: Defense against the Vulnerability". In: *Proceedings of the 8th Australasian Conference on Information Security and Privacy*. ACISP'03. Berlin, Heidelberg: Springer-Verlag, 2003, 226–236. DOI: 10.1007/3-540-45067-X_20 (see p. 41)

[132] N. P. SIONG and H. TOIVONEN. *Mee Too Crypto*. URL: https://gitlab.com/m2crypto/m2crypto (see p. 47)

[133] R. SNADER and N. BORISOV. "A Tune-up for Tor: Improving Security and Performance in the Tor Network". In: *Network and Distributed Security Symposium (NDSS 2008)*. Vol. 8. Internet Society, Feb. 2008 (see p. 81)

[134] R. SNADER and N. BORISOV. "EigenSpeed: Secure Peer-to-peer Bandwidth Evaluation". In: *Proceedings of the 8th international conference on Peer-to-peer systems*. IPTPS'09. Berkeley, CA, USA: USENIX Association, 2009, 9–9 (see p. 81)

[135] A. K. SOOD and S. ZEADALLY. A Taxonomy of Domain-Generation Algorithms. In: *IEEE Security Privacy*, **14**:4 (July 2016), 46–53. DOI: 10.1109/MSP.2016.76 (see p. 25)

[136] R. P. STANLEY. *Topics in Algebraic Combinatorics*. Course notes for Mathematics 192 (Algebraic Combinatorics), Harvard University, Cambridge U.S.A. 2000 (see pp. 109–111)

[137] I. STEINWART and A. CHRISTMANN. *Support Vector Machines*. Information Science and Statistics. Springer, 2008. DOI: 10.1007/978-0-387-77242-4 (see pp. 26, 28)

[138] A. SWARTZ. *Squaring the triangle: Secure, decentralized, human-readable names*. Jan. 2011. URL: http://www.aaronsw.com/weblog/squarezooko (see p. 104)

[139] P. SYVERSON. "Why I'm Not an Entropist". In: *Security Protocols XVII: 17th International Workshop, Cambridge, UK, April 1-3, 2009. Revised Selected Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, 213–230. DOI: 10.1007/978-3-642-36213-2_25 (see p. 19)

[140] P. SYVERSON, G. TSUDIK, M. REED, and C. LANDWEHR. "Towards an Analysis of Onion Routing Security". In: *International workshop on Designing privacy enhancing technologies: design issues in anonymity and unobservability*. New York, NY, USA: Springer-Verlag New York, Inc., 2001, 96–114 (see pp. 52, 53, 82)

[141] M. TARASIEWICZ and A. NEWMAN. Cryptocurrencies as Distributed Community Experiments. In: *Handbook of Digital Currency*, (2015), 201–222. DOI: 10.1016/b978-0-12-802117-0.00010-2 (see p. 1)

[142] *Team Cymru IP to ASN Lookup v1.0*. URL: https://whois.cymru.com (see p. 31)

[143] THE OPENSSL PROJECT. *OpenSSL: The Open Source toolkit for SSL/TLS*. URL: https://www.openssl.org (see pp. 47, 98)

[144] J. P. TIMPANARO, I. CHRISMENT, and O. FESTOR. "I2P's Usage Characterization". In: *Proceedings of the 4th International Conference on Traffic Monitoring and Analysis*. TMA'12. Berlin, Heidelberg: Springer-Verlag, 2012, 48–51. DOI: 10.1007/978-3-642-28534-9_5 (see pp. 1, 106)

[145] C. TRONCOSO, B. GIERLICHS, B. PRENEEL, and I. VERBAUWHEDE. "Perfect Matching Disclosure Attacks". In: *Proceedings of the 8th international symposium on Privacy Enhancing Technologies*. PETS'08. Berlin, Heidelberg: Springer-Verlag, 2008, 2–23. DOI: 10.1007/978-3-540-70630-4_2 (see p. 81)

[146] R. VAN RIJSWIJK-DEIJ, A. SPEROTTO, and A. PRAS. "DNSSEC and its potential for DDoS attacks". In: *Proceedings of the 2014 Conference on Internet Measurement Conference*. IMC '14. New York, NY, USA: Association for Computing Machinery (ACM), 2014, 449–460. DOI: 10.1145/2663716.2663731 (see p. 105)

[147] R. VAN RIJSWIJK-DEIJ, A. SPEROTTO, and A. PRAS. Making the Case for Elliptic Curves in DNSSEC. In: *ACM SIGCOMM Computer Communication Review*, **45**:5 (Sept. 2015), 13–19. DOI: 10.1145/2831347.2831350 (see p. 105)

[148] M. WACHS, M. SCHANZENBACH, and C. GROTHOFF. "A Censorship-Resistant, Privacy-Enhancing and Fully Decentralized Name System". In: *Proceedings of the 13th International Conference on Cryptology and Network Security - Volume 8813*. New York, NY, USA: Springer-Verlag New York, Inc., 2014, 127–142. DOI: 10.1007/978-3-319-12280-9_9 (see p. 104)

[149] C. WANG, D. SHI, and X. XU. AIB-OR: Improving Onion Routing Circuit Construction Using Anonymous Identity-Based Cryptosystems. In: *PLOS ONE*, **10**:3 (Mar. 2015). Ed. by G. XIAO. DOI: 10.1371/journal.pone.0121226 (see p. 88)

[150] T. WANG, K. BAUER, C. FORERO, and I. GOLDBERG. *Congestion-aware Path Selection for Tor*. Tech. rep. 20. University of Waterloo, 2011 (see p. 82)

[151] T. WANG, K. BAUER, C. FORERO, and I. GOLDBERG. *Congestion-Aware Path Selection for Tor*. In: *16th International Conference on Financial Cryptography and Data Security (FC'12)*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, 98–113. DOI: 10.1007/978-3-642-32946-3_9 (see p. 82)

[152] M. WRIGHT, M. ADLER, B. N. LEVINE, and C. SHIELDS. "An Analysis of the Degradation of Anonymous Protocols". In: *Proceedings of ISOC Symposium Network and Distributed System Security (NDSS)*. Feb. 2002, 38–50 (see p. 40)

[153] M. K. WRIGHT, M. ADLER, B. N. LEVINE, and C. SHIELDS. Passive-Logging Attacks Against Anonymous Communications Systems. In: *ACM Transactions on Information and System Security*, **11**:2 (May 2008), 1–34. DOI: 10.1145/1330332.1330335 (see p. 40)

[154] J. WU, L. ZHANG, J. LIANG, S. QU, and Z. NI. "A Comparative Study for Fast-Flux Service Networks Detection". In: *The 6th International Conference on Networked Computing and Advanced Information Management*. Aug. 2010, 346–350 (see p. 3)

[155] Y. XU, Y. LU, and Z. GUO. "The Availability of Fast-Flux Service Networks". In: *Mobile and Wireless Networking (iCOST), 2011 International Conference on Selected Topics in Mobile & Wireless Networking*. Institute of Electrical & Electronics Engineers (IEEE), Oct. 2011, 89–93. DOI: 10.1109/iCOST.2011.6085842 (see p. 26)

[156] L. YANG and F. LI. "mTor: A multipath Tor routing beyond bandwidth throttling". In: *2015 IEEE Conference on Communications and Network Security (CNS)*. Institute of Electrical and Electronics Engineers (IEEE), Sept. 2015, 479–487. DOI: 10.1109/CNS.2015.7346860 (see p. 106)

[157] S. YEKHANIN. Private Information Retrieval. In: *Communications of the ACM*, **53**:4 (Apr. 2010), 68–73. DOI: 10.1145/1721654.1721674 (see p. 2)

[158] X. YU, B. ZHANG, L. KANG, and J. CHEN. Fast-Flux Botnet Detection Based on Weighted SVM. In: *Information Technology Journal*, **11**:8 (Aug. 2012), 1048–1055. DOI: 10.3923/itj.2012.1048.1055 (see p. 26)

[159] N. ZHANG, W. YU, X. FU, and S. DAS. "gPath: A Game-Theoretic Path Selection Algorithm to Protect Tor's Anonymity". In: *Decision and Game Theory for Security*. Ed. by T. ALPCAN, L. BUTTYÁN, and J. BARAS. Vol. 6442. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, 58–71. DOI: 10.1007/978-3-642-17197-0_4 (see p. 81)

[160] F. ZHAO, Y. HORI, and K. SAKURAI. "Analysis of Privacy Disclosure in DNS Query". In: *Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering*. MUE '07. Washington, DC, USA: IEEE Computer Society, 2007, 952–957. DOI: 10.1109/MUE.2007.84 (see pp. 14, 38, 39, 41–43)

[161] F. ZHAO, Y. HORI, and K. SAKURAI. "Two-Servers PIR Based DNS Query Scheme with Privacy-Preserving". In: *Proceedings of the The 2007 International Conference on Intelligent Pervasive Computing*. IPC '07. Washington, DC, USA: IEEE Computer Society, Oct. 2007, 299–302. DOI: 10.1109/IPC.2007.107 (see pp. 38, 39, 42, 43)

[162] C. V. ZHOU, C. LECKIE, and S. KARUNASEKERA. Collaborative Detection of Fast Flux Phishing Domains. In: *Journal of Networks*, **4**:1 (Feb. 2009), 75–84. DOI: 10.4304/jnw.4.1.75-84 (see p. 26)

[163] C. V. ZHOU, C. LECKIE, and S. KARUNASEKERA. A Survey of Coordinated Attacks and Collaborative Intrusion Detection. In: *Computers & Security*, **29**:1 (Feb. 2010), 124–140. DOI: 10.1016/j.cose.2009.06.008 (see p. 26)

[164] C. V. ZHOU, C. LECKIE, S. KARUNASEKERA, and T. PENG. "A Self-Healing, Self-Protecting Collaborative Intrusion Detection Architecture to Trace-Back Fast-Flux Phishing Domains". In: *NOMS Workshops 2008 - IEEE Network Operations and Management Symposium Workshops*. Institute of Electrical & Electronics Engineers (IEEE), Apr. 2008, 321–327. DOI: 10.1109/NOMSW.2007.50 (see p. 26)

[165] Y.-L. ZHOU, Q.-S. LI, Q. MIAO, and K. YIM. DGA-Based Botnet Detection Using DNS Traffic. In: *Journal of Internet Services and Information Security (JISIS)*, **3**:3/4 (2013), 116–123 (see p. 105)