POLITECNICO DI MILANO

DEPARTMENT OF ELECTRONICS, INFORMATION AND BIOENGINEERING

DOCTORAL PROGRAMME IN INFORMATION TECHNOLOGY - TELECOMMUNICATIONS

Privacy-Preserving Service Delivery in Internet

Doctoral Dissertation of:
Davide Andreoletti

Supervisor:
Prof. Massimo Tornatore

Coadvisor:
Prof. Silvia Giordano

Tutor:
Prof. Matteo Cesana

Supervisor of the Doctoral Program:
Prof. Andrea Bonarini

2019 – Cycle XXXII

Acknowledgments

Firsty, I would like to sincerely thank my supervisors Prof. Massimo Tornatore from Politecnico di Milano and Prof. Silvia Giordano from SUPSI. I especially thank them for always giving me the freedom to explore the topics that interested me the most, even when they did not seem too much promising. The help they gave me to turn my intuitions into rigoruous research has been of invaluable importance for me. Besides my supervisors, my sincere gratitude also goes to Prof. Giacomo Verticale who, with his immense knowledge, has increadibly helped me to see problems from multiple perspectives, and, in general, to broaden my horizons in research. I also want to thank Dr. Cristina Rottondi and my friends and collegues Omran, Seb and Luca, with whom I had the pleasure to conduct several works and, more broadly, for the stimulating discussions we have had. The support of all the above people has significantly contributed to make the writing of this Thesis possible. I thank all my colleagues in SUPSI, and especially Alan for the (almost) daily debates about the (unproven) swiss superiority. Such discussions definitly made my temperament stronger (and more patient toward biased arguments). My gratidute also goes to all the colleagues in Politecnico (Mem, Ligia, Leila, GrandeMario, Prof. Francesco Musumeci, Prof. Guido Maier, etc...). They have always been friendly to me, and the time spent together during this years has been increadibly enjoyable. Another special thank goes to all my lifelong friends, who have always been of great moral support to me, and to my two flatmates Teresa and Luisa. I really appreciated their continuous support, friendliness and openness, and I mostly thank them for not complaining (at least not too much!) about my unconventional way of keeping the house in order. Then, I am greatful to

Tanya, who I had the priviledge to supervise during her master thesis at SUPSI, both for her very nice work and for the really good moments spent together. I also thank Tracy, whose intelligence and depth always help me to keep a different and open view on life. Last but not the least, I would like to thank my family for supporting me throughout the choices I take in my life.

Abstract

Today's Internet is a complex system through which users are provided with the most varied services. Internet is composed of several different and (generally) independent entities, such as Internet Service Providers (ISPs), Content Providers (CPs) and Online Social Network Providers (OSNs) that cooperate to deliver services to final users. For instance, ISPs deliver to final users the Internet traffic that carries the contents (e.g., a movie) that is originally hosted in a data center owned by a CP. By exploiting several enabling technologies (e.g., in-network caching) and the different information the involved entities have (e.g., about the final users and the network infrastructures), this cooperation can be taken even further to improve the QoE perceived by users.

For instance, the ISP can maintain cache servers within its network and make the CP remotely manage them (e.g., by selecting the most popular contents worth caching), thus reducing contents' retrieval latency and network congestion probability. Whilst being beneficial to all the involved parties, such improved cooperation may require the exposition of sensitive and business-critical information (e.g., about network infrastructure) that raises severe privacy concerns. The overall objective of this research is the development of methodologies to enable the main Internet players to cooperate and exchange information for realizing improved services while fulfilling their privacy requirements.

In general terms, guaranteeing privacy comes at the expenses of service effectiveness degradation and/or at the cost of introducing a non-negligible overhead of data exchanged between the cooperating parties. In this thesis, we propose feasible and readily applicable privacy-preserving solutions for several Internet-based services, such as video content delivery and online social networking.

Firstly, we focus on the application of video contents' caching strategies jointly performed by ISPs and CPs. We design privacy-preserving protocols based on data perturbation and secure multiparty computation to ensure caching effectiveness (e.g., maximization of hit-ratio and minimization of retrieval latency) while guaranteeing that sensitive information are not disclosed (e.g., contents' popularity, users' requests and locations are only known to the legitimate party). Then, we also propose a protocol based on Shamir Secret Sharing (SSS) to realize caching strategies that are both privacy-preserving and compliant with Network Neutrality principles.

Moreover, we propose a machine-learning-based tool that Twitter users can employ to measure the vulnerability to attacks aimed at inferring their location from publicly-available data. This tool

also allows to quantitatively evaluate the effects that several factors (e.g., the frequency of exposition of location data) have on users' privacy, thus enabling their proper control.

Finally, we study the problem of optimally deploying a virtual graph over a wide-area network composed of several independent and mutually-distrustful ISPs. We develop a reinforcement learning algorithm based on SSS which is capable to effectively deploy the virtual graph while not requiring the exposition of salient infrastructural information (e.g., cost of embedding into the physical nodes).

List of Publications

P1 **D. Andreoletti**, O. Ayoub, S. Giordano, G. Verticale and M. Tornatore, *"Privacy-Preserving Caching in ISP Networks"*. In *proceedings of IEEE International Conference on High Performance Switching and Routing (HPSR)*, pp. 1-6, May 2019.

The material of this publication contributes to Chapter 2.

P2 **D. Andreoletti**, S. Giordano, C. Rottondi, M. Tornatore and G. Verticale, *"To be neutral or not neutral? The in-network caching dilemma"*. In *IEEE Internet Computing*, 22(6), 18-26.

The material of this publication contributes to Chapter 3.

P3 **D. Andreoletti**, C. Rottondi, S. Giordano, G. Verticale and M. Tornatore, *"An open privacy-preserving and scalable protocol for a Network-Neutrality compliant caching"*. In *proceedings of IEEE International Conference on Communications*, Shanghai, pp. 1-6, May, 2019.

The material of this publication contributes to Chapter 4.

P4 **D. Andreoletti**, O. Ayoub, C. Rottondi, S. Giordano, G. Verticale and M. Tornatore, *"A Privacy-Preserving Protocol for Network-Neutral Caching in ISP Networks"*. Accepted for publication at *IEEE Access*.

The material of this publication contributes to Chapter 4.

P5 **D. Andreoletti**, S. Giordano, G. Verticale and M. Tornatore, *"Discovering the Geographic Distribution of Live Videos' Users: A Privacy-Preserving Approach"*. In *proceedings of IEEE Global Communications Conference*, Abu Dhabi, pp. 1-6, December 2018.

The material of this publication contributes to Chapter 5.

P6 **D. Andreoletti**, L. Luceri, M. Tornatore, T. Braun and S. Giordano, *"Measurement and Control of Geo-Location Privacy on Twitter"*. Under submission at *Online Social Networks and Media*

The material of this publication contributes to Chapter 6.

P7 **D. Andreoletti**, T. Velichkova, G. Verticale, M. Tornatore and S. Giordano, *"A Privacy-Preserving Reinforcement Learning Algorithm for Virtual Network Embedding over a multiple-domain infrastructure"*. Under submission at *IEEE Transaction on Network and Service Management*.

The material of this publication contributes to Chapter 7.

Additional contributions that were not included in this Doctoral Dissertation

P8 O. Ayoub, F. Musumeci, **D. Andreoletti**, M. Mussini, M. Tornatore and A. Pattavina, *"Optimal Cache Deployment for Video-on-Demand Delivery in Optical Metro-Area Networks"*. In *proceedings of IEEE Global Communications Conference*, Abu Dhabi, pp. 1-6, December 2018.

P9 **D. Andreoletti**, S. Troia, F. Musumeci, S. Giordano, G. Maier and M. Tornatore, *Network Traffic Prediction based on Diffusion Convolutional Recurrent Neural Networks*. In *proceedings of IEEE International Workshop on Network Intelligence at INFOCOM*, Paris, pp. 1-6, April 2019.

P10 E. Frumento, F. Freschi, **D. Andreoletti** and A. Consoli, *"Victim Communication Stack (VCS): A flexible model to select the Human Attack Vector"*. In *proceedings of International Conference on Availability, Reliability and Security*, Reggio Calabria, pp. 1-6, September 2017.

Contents

Bibliography **141**

List of Figures

List of Tables

Introduction 1

Initially designed as a global network mainly aimed at connecting geographically-distributed computers, the Internet has rapidly become a complex infrastructure through which end-users are provided with a vast number of indispensable services. As a result of this transformation, today's Internet is a composition of many entities, which cooperate to perform the delivery of a service in a mutually-profitable manner. To better figure out which are the main entities in Internet and to understand their interdependency, let us think of a pyramidal structure where the basis is represented by the Internet Service Providers (ISPs), which own and manage the infrastructures that transport the Internet traffic. This traffic carries the services offered by the entities referred to as Over-The-Top (OTTs). Two remarkable examples of OTTs that are nowadays extremely popular are the Video Content Providers (CPs), which own and manage catalogues of video contents that users can retrieve, and the Online Social Network (OSN) providers, which offer users digital platforms where they can connect to each other and publish the most varied contents. On top of them, we find service providers that reach their users exploiting the platforms offered by giant OTTs (e.g., a Location-Based Service that performs advertising on a OSN).

We observe that the characteristics of the today's Internet open the door to the implementation of more complex cooperative schemes with respect to the basic ones that we have briefly mentioned. In particular, we notice that such entities (i) generally have different footprints (e.g., an ISP offers Internet connectivity within a limited area, while a CP distributes its contents on a global basis) and (ii) they possess different information about the final users (e.g., only the CPs know users' preferences). Moreover, (iii) by decoupling a service from the underlying physical devices, virtualization strategies are making the boundaries in Internet increasingly blurred (e.g., a service can be offered within an area not covered by its provider).

An example of service that particularly benefits from an increased cooperation (namely, between

ISPs and CPs) is video content distribution. In fact, if the CPs can serve their videos from network positions closer to the users (e.g., inside the ISP's network), the offered Quality of Experience (QoE) is enhanced, by virtue of a reduction of the retrieval latency and the congestion probability. For example, in one of the works described in this thesis, we address the problem of optimally deploying Virtual Servers (VSs) inside the network of the ISP, from which the CP can better serve Live Videos (LVs) to its viewers. This is a clear example of an optimization process that has to be executed jointly by multiple entities, as each of them possesses a portion of the data needed to perform the optimal deployment. Specifically, only the ISP knows the precise users' position (since it covers the last segment of traffic delivery) and, under the use of encryption schemes (which is a common practice nowadays), only the CP knows the LVs that users request. However, these data are deemed privacy-sensitive by users and considered business-critical assets by the owner entities, which may therefore not be willing (or entitled) to freely share them with each other.

To address this issue, in this thesis we develop methodologies aimed to improve the effectiveness of service delivery without sacrificing privacy. In other words, we propose *privacy-preserving* data sharing solutions approaches that make the entities involved in the delivery of a service only able to extract from these data the information that they need to improve the service, but not to violate privacy. To this end, we consider several services provided over the Internet, and we formally define the privacy requirements of the involved entities (e.g., CPs, ISPs and final users). As the privacy-preserving strategies often reduce the knowledge that it is possible to extract from data, it may happen that privacy can only be guaranteed at the cost of services' effectiveness degradation. In this research, we perform numerical evaluations of this phenomenon (which is better known as *privacy-utility* trade-off) in several scenarios.

In this research, the service that we mostly consider is video content distribution, which poses nowadays the strongest pressure on ISPs' infrastructures. Therefore, ISPs are constantly looking for innovative and reliable strategies to handle the impressive amount of traffic generated by the CPs. As explained in the aforementioned example, the ISP is in the favourable position to be much more than a simple traffic transportation carrier. For example, caching strategies represent a consolidated solution to both reduce network resource occupation in ISPs' networks and increase the QoE offered by the CPs. In fact, by performing caching an ISP stores a portion of the CPs' catalogues in servers located within its area (i.e., the *caches*) and serve contents directly from there. The result is a reduction of network traffic, retrieval latency and congestion probability.

To be effective, caching requires that the most requested (i.e., popular) contents are delivered from inside the ISP's network. In a context of all-encrypted web, however, the ISPs are not aware of the contents traversing their network and, therefore, they are not able to assess their popularity without implementing advanced forms of cooperation with the CPs. In relation to this, in this research we mainly consider the privacy issues resulting from such cooperation. For example, we develop strategies to guarantee that an ISP can perform effective caching without discovering the popularity of CP's contents, or we propose techniques to enable a CP to serve a LV from a source

close to its viewers, without knowing their position.

Then, we observe that the prioritization of traffic treatment achieved by implementing caching strategies should be carefully analyzed under the lens of Network Neutrality (NN) regulations. Specifically, we explore the problem of subdividing a limited ISPs' cache storage among a set of CPs in a way that is both efficient (i.e., that minimizes the network resource occupation) and fair towards the CPs that exploit the caching system. We propose our definition of NN-compliant caching and design a protocol to enforce it in a privacy-preserving manner (e.g., CPs are not required to expose the information about their contents' popularity).

Our attention is then shifted to the protection of users' location privacy in OSNs. Specifically, we develop a privacy awareness tool that Twitter's users can employ to (i) estimate how accurately their locations can be inferred from publicly-available data and (ii) to understand the factors that mainly affect their vulnerability to this inference. We also propose data perturbation techniques and provide a qualitative evaluation of the trade-off between users' privacy and effectiveness of a Location Based Service (LBS).

Finally, we focus on the privacy-preserving Virtual Network Embedding (VNE) problem over a multi-ISPs infrastructure. In the considered problem, a customer is willing to find the most cost-effective deployment of a set of virtual functions over a wide-network composed of several ISPs, which keep the information about their network infrastructure private (e.g., the cost of traversing a link is not exposed). In particular, we propose an algorithm based on Reinforcement Learning that is implemented using the Shamir Secret Sharing scheme.

1.1 Privacy-Preserving Caching

The increase of Internet traffic is mainly driven by the huge escalation of online streaming of video contents offered by CPs. As a matter of fact, the IP video traffic is expected to be the 82% of the overall IP traffic by 2022 [34]. For this reason, ISPs are always looking for reliable solutions to reduce the strong pressure that such traffic poses on their network infrastructures. In this respect, caching strategies represent a natural approach to address this problem, as serving portion of CPs' contents directly from the area of the ISP reduces the volume of traffic traversing the networks' links.

The effectiveness of caching mainly relies on the characteristics of video contents' popularity distributions. Remarkably, the popularity distribution of a video catalogue is characterized by a long-tail, i.e., few contents are, on average, requested much more than all the others. This specific property, which is widely-modeled with the Zipf distribution [25], allows ISPs to achieve a significant reduction of traffic traversing their networks by caching only the most popular contents, which represent a small fraction of the entire CPs' catalogues.

As commonly done in the literature, in this research we consider ISPs and CPs to be separate and independent entities, which jointly benefit from the application of caching strategies inside the network of the ISP. By applying caching, in fact, the former experience a reduction of traffic

burden inside their infrastructures, while the latter can guarantee an increased QoE to their users. However, CPs are not willing to disclose the information on their contents' popularity, which is required to perform an effective caching. In the following, we describe the main approaches that ISPs and CPs can follow to apply caching while guaranteeing CPs' privacy requirements. To our knowledge, *privacy-preserving* caching within ISPs' networks can be implemented following three main approaches:

- the ISP owns the caching system and exploits it to implement caching procedures (e.g., by directly selecting and serving the cached contents)
- the CP owns and manages the caching system, which is located within the area of the ISP
- the ISP owns a caching system that is remotely managed by the CP (e.g., the CP selects the content worth caching)

By adopting the first approach, ISPs have the highest control on the traffic generated by caching; for instance, they can decide in which cache servers contents have to be stored to achieve a specific traffic engineering objective [97]. Traditionally, this approach is applied by employing transparent caching strategies, by which an ISP analyzes the traffic traversing its network to infer the most requested contents. However, if contents are encrypted by CPs for security and privacy reasons (e.g., to protect the information about their contents' popularity), this approach can only be implemented if the ISP cooperates with the CPs. Recently, the authors of [115] propose a cooperative architecture to enable the application of efficient caching strategies of encrypted contents. From a high-level standpoint, this architecture works as follows: the CP associates its contents with pseudonyms that the ISP can analyze to infer contents' popularity without decrypting the contents themselves. In Chapter 2, we elaborate on privacy issues resulting from the employment of this architecture in a real scenario of cooperation between ISPs and CPs, showing that there exists a trade-off between CPs' privacy requirements and caching effectiveness.

The second approach is typically applied by big CPs, which have the scale to negotiate with ISPs such type of agreement (e.g., see the OpenConnect[1] program implemented by Netflix) to enhance the QoE experienced by their users. In Chapter 3 we elaborate on possible issues that this cooperative scheme raises in relation to Network Neutrality principles. Notice that the implementation of this cooperative scheme does not pose privacy issue, since all the caching process (i.e., selection of the contents worth storing and content delivery) is executed by the CP itself (hence, the ISP is not able to obtain any information about the popularity of CPs' contents).

The third approach allows ISPs to partition their caching resources among the CPs, which remotely manage the slice assigned to them. This approach is the most widely-employed to perform caching nowadays and several strategies have been proposed to subdivide the caching storage among

[1] https://openconnect.netflix.com/

the CPs in such a way that the resulting caching is effective for the ISP and privacy-preserving for the CPs. For example, the authors of [9] propose a strategy to assign to CPs portions of storage proportional to the Caching Hit-Ratio that each of them experiences. By following this approach, the ISP only obtains aggregated information about CPs' contents' popularity, and can therefore be considered privacy-preserving. In Chapter 4 we describe our solution to make the ISP able to assign CPs portion of caches' storage proportional to the popularity of their contents, while not requiring the exposition of this sensitive information.

1.2 Network Neutrality

According to many, the Internet can foster social and economic growth only if it provides open and nondiscriminatory access to information. Such view does not really describe current Internet behaviour but can be regarded as an ideal goal, which Network Neutrality (NN) regulations try to achieve. In practice, Internet evolution resulted in a complex service-delivery chain where ISPs own and manage the infrastructures that CPs exploit to offer services to their end-users. As providers of new and increasingly advanced services, CPs play a key role in fulfilling the promise of an economically valuable Internet. Hence, in a scenario of interaction between these two actors, the ISPs must upgrade their infrastructure to avoid becoming the bottleneck of the entire process.

In fact, the development of an efficient Internet infrastructure would result in a virtuous circle in which CPs can offer their services with high quality of service (QoS) guarantees and, in this way, induce an increasing number of users to subscribe with ISPs. However, this process is hindered by some practical issues: as vendors of valuable services, the CPs take the lion's share, while the ISPs face the risk of becoming simple providers of connectivity. This unbalance in revenue distribution risks to jeopardize the ISPs, as they are downgraded to commodity providers, and it is exacerbated by the purest NN vision, according to which ISPs should have limited or no control on the traffic traversing their network. However, this *neutral* delivery paradigm is not suitable anymore in a context where service requirements are becoming heterogeneous to such a degree that they necessarily require a different treatment of traffic. For example, it is undoubtable that users desire more guarantees for the performance of video on demand (VoD) that for email exchange. Considering next-generation networks (e.g., 5G), the current trend is to accommodate, over the same physical infrastructure, several virtual networks specifically tailored to run various bandwidth-hungry services. For example, the network slicing [102] paradigm embodies the principle that ISPs reserve to an external entity dedicated resources that fulfil given requirements (e.g., in terms of latency). This is a form of traffic differentiation and raises questions on the effective neutrality of ISPs. Given their social and economic role in deploying Internet infrastructures, should ISPs have the right to decide how to treat traffic in their network? If yes, to what extent can this be done neutrally and consistently with the current requirements of today's services?

The debate on NN is a long-standing one. Several frameworks have been proposed with the aim

of reaching a common definition of neutrality in an era where traffic differentiation is mandatory. The proponents of pure NN see ISPs are mere *pipes* that should be agnostic to the contents they carry. On the other hand, NN opponents would give ISPs the greatest control as a legitimate action to increase their revenue and consequently foster innovation. The reality of today's legislation is more nuanced and proposes different frameworks that aim at balancing the inherent trade-off between NN and QoS. The philosophy of such frameworks, for a thorough description of which we refer the reader to reference [113] can be summarized with the following statement: traffic differentiation should be allowed as long as it is not discriminatory for the CPs. Under such lens, for example, traffic can be categorized in classes (e.g., Video On Demand or VoIP) and differentiated accordingly, but contents belonging to the same class cannot be discriminated based on the owner CP. A complete review of currently-available strategies to detect illicit traffic discrimination procedures (i.e., not compliant with NN) can be found at reference [54].

In this research, we discuss a topic that, perhaps surprisingly, has rarely been treated under the lens of NN, i.e., in-network caching. As previously described, in-network caching is the process by which ISPs store in their networks the most popular contents to reduce traffic coming from external systems, i.e., CPs. By using this strategy, contents are retrieved from closer servers and users experience a superior QoE. However, because of the limited storage of caches, in-network caching is an intrinsically selective process and, as such, raises discriminatory concerns. How can we perform caching to avoid that neither CPs nor users are discriminated? In Chapter 3 we provide a possible definition of NN-compliant caching as the process according to which an ISP allocates to the CPs a portion of cache storage proportional to the popularity of their contents. In this way, forms of arbitrary agreements between ISPs and CPs are avoided, as each CP is treated by the ISP only based on its attractiveness towards the users. However, to enforce this definition the ISP has to obtain the information about CPs' contents popularity, which is unavailable if encryption schemes are applied. To address this issue, we then present in Chapter 4 an open protocol that ISPs and CPs can employ to compute such popularity-based subdivision in a privacy-preserving manner (i.e., without requiring the decryption of CPs' contents and the exposition of CPs' contents popularity).

1.3 Privacy-Preserving Data Sharing

In general terms, privacy is related to the amount of knowledge that can be extracted by the use of an information source. Hence, although being a rather application-dependent concept, protecting privacy means keeping the information gain below a predefined threshold [106]. As previously mentioned, the entities involved in service delivery in Internet possess different data, e.g., about the final users and network infrastructures. The sharing of these data would lead to an improved and more user-tailored service, but comes with privacy concerns. In this Section, we describe the main existing privacy-preserving approaches that we employ in our research.

1.3.1 Shamir Secret Sharing

A (W, T) Shamir Secret Sharing (SSS) scheme [108] is a cryptographic technique that allows to share a secret s among a set of W participants in such a way that its reconstruction can only be performed by the collusion of any subset of at least T participants. We use the notation $[\![s]\!]_P$ to indicate the share of s assigned to the participant P. The SSS is based on the principle that any polynomial of degree $T-1$ can be perfectly reconstructed from the knowledge of T points that it intercepts. Let $s \in \mathbb{Z}_q$ be the secret (with q a prime number greater than all the possible secrets) and let $a_1, a_2, ..., a_{T-1}$ be the coefficients of the polynomial, which are random integers uniformly distributed in $[0, q-1]$. The participant P receives $[\![s]\!]_P = (x_P, y_P)$, with x_P being an integer number (distinct for each participant) and $y_P = s + a_1 x_P +_2 x_P^2 + ... + a_{T-1} x_P^{T-1}$. The reconstruction of s can be performed by means of interpolation algorithms, e.g., the Lagrange interpolation.

SSS has homomorphic properties, i.e., it is possible to perform several operations on the shares that result on the same operations performed on the secrets. For example, SSS is natively homomorphic with respect to the addition. In practice, a participant P owning the shares of two secret numbers (say $[\![s_1]\!]_P$ and $[\![s_2]\!]_P$), can obtain the share $[\![s_3]\!]_P$ corresponding to the secret $s_3 = s_1 + s_2$ by simply summing its shares ($[\![s_1]\!]_P + [\![s_2]\!]_P$). More in general, a participant can compute any linear combination of its shares.

However, the SSS is not homomorphic with respect to the multiplication (i.e., given the shares of two secrets s_1 and s_2, $[\![s_1 \cdot s_2]\!] \neq [\![s_1]\!] \cdot [\![s_2]\!]$). As the multiplication is extremely important to build complex operations, we address this issue by exploiting the multiplication scheme proposed in [19]. This scheme requires the parties involved in the multiplication to share with each other a multiplicative triple $[\![a]\!], [\![b]\!], [\![c]\!]$ such that $a \cdot b = c$. The security of the multiplication scheme described in [19] is based on the assumption that none of the involved parties is able to obtain the secrets a, b, c from the relative shares $[\![a]\!], [\![b]\!], [\![c]\!]$. The shares of the multiplication triple can be pre-computed in a secure manner using the scheme proposed in [37].

In our research, we build privacy-preserving protocols under the SSS, which are based on both secure addition and multiplication. For example, in Chapter 4 we propose a protocol to enable an ISP compute a NN-compliant subdivision of its cache storage. As described in the previous Section, enforcing such subdivision means to assign to a CP that exploits the caching system a portion of storage proportional to the popularity of its contents. By exploiting the homomorphic properties of SSS, this subdivision can be computed by performing operations on shares, thus keeping private data that the CPs do not want to expose (i.e., the popularity of their contents).

In Chapter 7, we then describe our implementation of a Reinforcement Learning algorithm proposed to perform the optimal deployment of a virtual network over physical infrastructures owned by multiple ISPs. This optimization requires the involved ISPs to share with each other business-critical information about their infrastructure (e.g., the cost of traversing their links). Our approach is based on computation performed over the shares relative to the secrets that ISPs do not

disclose.

As briefly described in the beginning of this subsection, SSS is based on the association of a secret with a polynomial. This fact reduces the computational efficiency with respect to simpler secret sharing schemes (e.g., trivial secret sharing). However, our proposed privacy-preserving operators are build upon cryptographic primitives designed under the SSS scheme (e.g.. the aforementioned multiplication protocol). We leave the adaptation of such primitives to simpler sharing schemes as a future research direction. In particular, we plan to evaluate the gain (e.g., in terms of reduced computational time and exchanged data overhead) given by such lighter schemes.

1.3.2 Secure Multiple-Party Computation

A secure multiply-party computation method aims to enable the computation of a function over a set of data owned by several mutually-distrustful parties that are not willing to disclose them. A typical example of application consists in 2 people who want to discover who is the richest, but do not want to expose their income. In Chapter 5, we consider the problem of deploying Virtual Servers (VSs) within the area of an ISP, from which a CP can better deliver live videos to its users. Assuming that encryption is applied, the ISP is not aware of how requests are geographically-distributed in its area. Therefore, the ISP is unable to deploy the VSs towards some optimization objective (e.g., to minimize the retrieval latency). In the considered case, the ISP has the information of the geographically distribution of the users (but it does not know the contents they request), while the CP knows the contents of users requests (but is not aware of their position). To address this problem, we employ an existing secure multiple-party computation protocol to make a CP and an ISP jointly compute this geographical distribution of requests without requiring the share of their data.

1.3.3 Data Perturbation

Data perturbation techniques add noise to data in such a way to reduce the knowledge that it is possible to obtain from their analysis. In this research, we make large use of such techniques: for example, we apply data perturbation strategies to increase the privacy of a CP that cooperates with an ISP to perform an optimized video contents delivery. We describe such strategies in Chapter 2 and in Chapter 5, where the objective is to protect users' confidentiality. Then, we also apply data perturbation strategies to protect the privacy of users' location in Online Social Networks. We describe such strategies in Chapter 6.

1.3.3.1 Trade-off between Utility and Privacy

Unlike the first two privacy-preserving strategies (namely, Shamir Secret Sharing and Secure Multiple Party Computation), data perturbation improves privacy at the expenses of the information obtainable from it. As a matter of fact, the perturbation of data induces a trade-off between the privacy and the knowledge that it is possible to extract from them (referred here to as *utility*) [106].

In this research, we explore the trade-off between privacy and utility in several contexts. For example, in Chapters 2 and 5 the utility is measured considering the effectiveness of the implementation of enhanced video content delivery strategies (e.g., privacy-preserving caching). In Chapter 6 the utility is the effectiveness of a Location-Based Service that exploits the information about users' location published on the Online Social Network Twitter.

1.4 Contribution and Thesis Outline

This PhD research work is summarized in the following main activities:

1- Proposal of privacy-preserving solutions to enable an effective cooperation of entities in Internet, mostly considering the service of video content delivery.

2- Proposal of a privacy awareness tool to measure and control the vulnerability of OSNs' users to attacks aimed at inferring their location from public data.

The first activity mainly focuses on the design of protocols aimed at allowing an enhanced video content caching, i.e., privacy-preserving and network neutrality compliant. For example, we propose protocols that allow ISPs and CPs to jointly perform caching while protecting CPs' contents popularity, users' location and requests and ISPs' infrastructural details (e.g., caches' dimension). Along this research line, we also propose an approach based on reinforcement learning that effectively deploys Virtual Graphs over a multi-domain infrastructure while not requiring the exposition of sensitive data (e.g., cost of embedding a virtual node into a physical one). The second activity considers the problem of protecting the location privacy of OSNs' users (and specifically, Twitter's users). We propose a novel deep learning methodology to infer unexposed users' location from the publicly-available ones. We also propose data perturbation strategies to increase users' privacy, that we successively model as a combination of several factors, such as users' mobility and level of data perturbation. More in detail, our research work is organized through the remainder of this thesis as follows:

- In Chapter 2 we consider privacy issues resulting from the employment of an architectural solution designed to enable the effective caching of encrypted contents. This architecture aims to allow an ISP to perform caching of video contents owned by a CP that is willing to maintain them encrypted to ensure the confidentiality of its users. By employing this architecture a CP associates to each of its contents a pseudonym that the ISP is allowed to read. By counting the occurrences of the pseudonyms, the ISP discovers if a content is sufficiently popular to be cached without decrypting it. We observe that, by counting the occurrences of the pseudonyms, the ISP can still obtain information that may threaten CPs' privacy. In particular, the ISP may be able to obtain valuable information about CPs' popularity patterns and, if provided with additional information (e.g., publicly-available hit-parades) can guess

with high probability the contents that users are requesting (therefore violating also their confidentiality). To address this issue, we first formalize a data perturbation strategy that consists in frequently replacing the pseudonym associated with each content. By doing so, the CP reduces the knowledge about contents' popularity that the ISP can obtain from analyzing the occurrences of the relative pseudonyms, thus also reducing the effectiveness of caching. We then formally define the privacy requirements of the CPs and we quantify the trade-off between privacy and caching effectiveness induced by the applied data perturbation. Specifically, we perform simulations to measure the effectiveness of caching considering hit-ratio, average retrieval latency and average traffic load on ISP's network links. Results show a linear decrease of caching performance with increasing the number of pseudonyms replacements. On the other hand, the improvement of privacy is superlinear with increasing this variable. Hence, there is an optimum number of replacements such that CPs' privacy is significantly increased while accepting a low deterioration of caching performance.

- In Chapter 3 we discuss the issue of caching in relation to Network Neutrality principles. Initially, we identify several characteristics of todays' Internet (especially, the wide use of contents' encryption) that make the cooperation between ISPs and CPs mandatory to implement caching strategies. Given this, we provide a quantitative assessment of the caching performance experienced by CPs based on the cooperative scheme implemented with an ISP. We conclude that the employed cooperative scheme may lead to a discriminatory treatment of several CPs, which should be prohibited in a NN-neutral scenario. Finally, we provide a possible definition of NN-compliant caching: in our view, an ISP performs a neutral caching if it allocates to each CP a portion of storage proportional to the popularity of its contents. This approach leads to the highest caching performance for the ISP and allows to avoid arbitrary forms of cooperation with CPs.

- In Chapter 4 we propose a protocol to enable an ISP to compute a NN-compliant subdivision of its cache storage in a privacy-preserving manner. In particular, we build a protocol based on the Shamir Secret Sharing scheme that allows to obtain such subdivision without requiring the ISP and the CPs to share with each other sensitive information (e.g., the available cache storage and the popularity of the contents). Firstly, we design a simplified version of the protocol, which can work under several assumptions (e.g., all the contents are required to be of the same size). We perform experiments characterized by the presence of several CPs and an ISP owning a single cache server. Results suggest that this subdivision leads to a significant improvement with respect to a static cache subdivision, where each CPs receive the same amount of cache storage. Moreover, the gain with respect to this baseline (measured considering the Hit-Rate) scales with increasing number of CPs and contents. We then propose an improved version of the protocol. In particular, we develop a protocol that allows to fully utilize the cache storage (which was not always guaranteed in the previous version) and does not require all the contents

to be of the same size. To confirm the benefits of using this protocol, we then perform further experiments considering an ISP owning a network of caches and evaluating additional metrics with respect to the Hit-Ratio. In particular, we show that the subdivision computed with our protocol leads to a significant reduction of ISP's network resource occupation with respect to the considered baselines. Finally, we evaluate the data overhead introduced by applying the protocol and conclude that it is acceptable considering the reduction of network resource occupation and the improvement of caching effectiveness achieved by using it.

- In Chapter 5 we consider the problem of deploying, within the network of an ISP, Virtual Servers (VSs) that a CP exploits to stream Live Videos (LVs) to its users. As the requests for LVs are more localized than traditional VoD [100], the knowledge of the geographic distribution of requests is crucial to perform an optimal deployment. In our work, we assume that both the ISP and the CP are willing to deploy the VSs to minimize the average number of network hops crossed by the LVs to reach their viewers. This optimization objective, in fact, leads to a reduction of network resource occupation (ISP's goal) and retrieval latency (CP's goal). Firstly, we observe that ISP and CP have complementary information about their users: the ISP knows the location of users within its area (as it delivers traffic to them) but not the content of their requests (which are encrypted); on the other hand, the CP knows users' requests but not their position. We then formally define the privacy requirements of the ISP and the CP, which prescribe that they cannot freely exchange the data they possess. To address this problem, we employ an existing secure multiple-party computation protocol that allows the ISP to obtain an aggregated information on the number of requests issued for each LV from a specific geographical area. Based on this information, the ISP performs the optimal deployment of the VSs. We compare the effectiveness of the deployment with a baseline approach. Results show that the average number of crossed hops is significantly reduced if the optimization is performed with the knowledge of the geographical distribution of the requests computed with our privacy-preserving protocol. We then notice that, while basic privacy requirements are fulfilled at no expenses of service effectiveness, more demanding privacy objectives are achieved only if the CP applies perturbation strategies to its data during the execution of the protocol. We evaluate the resulting trade-off between privacy and service effectiveness and conclude that highly-demanding privacy requirements can only be fulfilled at a significant reduction of service effectiveness.

- In Chapter 6 we consider the problem of how to measure and control the privacy leakage given by the public exposition of users' location (action referred to as contents' geo-tagging) on the Online Social Network Twitter. Specifically, we quantify users' level of privacy and show how to properly control the factors affecting it. We define privacy as the geographic distance between the actual location of a user and the one that can be inferred from publicly-available data (i.e., the geo-tags published by users within the OSN). To quantify users' privacy, we initially

propose a novel deep learning architecture that is trained to infer unexposed locations from the publicly-available ones. Results show that most of the geo-tags can be accurately inferred (60% with an error below 1 km). These alarming results drive us to propose two data perturbation strategies that users can employ to increase their privacy. Then, we model the privacy of a user as a combination of several characteristics, such as the frequency of her geo-tags, the employed data perturbation strategy and mobility-related features (e.g., average distance between two successive geo-tags). This model is learned using the Random Forest algorithm, which allows to quantify the impact that each feature has on the privacy of a user. We observe that privacy is mainly influenced by the mobility of a user and by the type and volume of data perturbation that she applies. Finally, we provide a qualitative analysis of the trade-off between users' privacy and effectiveness of a Location-Based Advertisement service that reaches its users on the OSN's platform. This trade-off is induced by the perturbation that users apply to their public geo-tags. The main outcome of the analysis is that users can significantly increase their privacy without highly reducing the effectiveness of the advertisement.

- In Chapter 7 we propose a Reinforcement Learning (RL) algorithm to perform Virtual Network Embedding (VNE) over a multi-domain infrastructure in a privacy-preserving manner. In this scenario, several independent and mutually-distrustful ISPs form a consortium that a customer exploits to provide a virtual service to its users. In particular, the customer is willing to effectively deploy a Virtual Graph (VG) over such multi-ISP infrastructure. However, the application of VNE algorithms in this scenario is hindered by the fact that, to protect their privacy, ISPs do not expose details of their infrastructure needed to perform an effective deployment. Following a common privacy-preserving approach, the embedding may be performed by the customer based on the abstract view of the multi-domain infrastructure that ISPs accept to expose, i.e., Limited Information Disclosure (LID). With this approach, embedding is sub-optimal (e.g., embedding cost is not minimized) in comparison with the case where all information is available, i.e., Full Information Disclosure (FID). We propose a Reinforcement-Learning-based algorithm able to process data that customer and ISPs cipher under the Shamir Secret Sharing (SSS) scheme. This approach guarantees total privacy to both customer and the ISPs (e.g., details about a virtual function are only revealed to the ISP in charge of hosting it) and achieves comparable embedding cost of an existing FID heuristic, as observed from extensive simulations. The main drawback of our algorithm is the high overhead of data that ISPs and customer need to exchange with each other to execute it. We explore the trade-off between embedding cost and data overhead resulting from reducing the number of expensive operations. Results show that intermediary embedding costs between the FID and LID heuristics can be obtained at a significant reduction of data overhead, while not sacrificing any privacy guarantees.

- Chapter 8 draws the conclusion of the thesis.

Privacy-Preserving Caching 2

Nowadays, CPs increasingly encrypt the traffic directed towards their users, thus achieving two main objectives: the protection of users' confidentiality and the protection of contents' popularity (i.e., the number of times that a content is requested). The latter, in particular, provides useful information about CPs' attractiveness in the market, and it is therefore considered a business-critical asset that has to be maintained private. By hiding contents' popularity, however, the CPs also make the ISP unable to perform in-network caching, which is effective when the most requested (i.e., popular) contents are cached. A recently-proposed architectural solution enables the application of an effective yet privacy-preserving caching in ISP networks. We observe that, using this architecture, there is a trade-off between CPs' privacy requirements and caching effectiveness (e.g., hit-ratio and retrieval latency). In this Chapter, we provide a formalization of privacy in this context and, by means of simulations over both synthetic and real data, we perform an evaluation of such trade-off.

2.1 Motivation

Among the services currently provided over the Internet, video content distribution is the one that poses the strongest pressure on Internet infrastructures [90]. In-network caching is widely-regarded as a simple yet effective solution to reduce this pressure, as ISPs can serve a portion of the CPs' contents directly from sources close to the end-users (i.e., the caches). In this way, the ISP significantly reduces the traffic burden within its network, while the CPs guarantee a superior Quality of Experience (QoE) to their users (e.g., because retrieval latency and congestion probability are reduced).

As mentioned in the previous Chapter, there are several approaches to manage a caching system. Just to briefly summarize them, (i) caches can be owned and managed by the ISP (i.e., the ISP selects the contents worth caching and directly serve them to users), (ii) caches can be owned by the

ISP and manged by the CP (i.e., the CP selects the contents to store on the caches and directly serve them to users) and (iii) caches can be owned and managed by the CP (i.e., caches are located within the area of the ISP and directly connected to its network infrastructure, but maintained and managed by the CPs). In this Chapter, we consider an ISP that owns a system of caches and implements the first approach, which gives ISPs the highest level of control of the caching process. For example, the ISP can select the contents to cache and where (i.e., in which caches) to store their replicas in order to achieve some optimization objectives, e.g., effective traffic engineering. In case encryption techniques were not applied, the ISP could perform *transparent caching*, i.e., analyze the contents traversing its network and, eventually, decide to cache them (e.g., if sufficiently popular). As nowadays encryption is widely used, the ISP cannot easily distinguish the contents and, therefore, it cannot count their occurrences to assess their popularity.

An architectural solution specifically-designed to enable an effective caching of encrypted contents has been proposed in [116]. From a high-level standpoint, this architecture allows a CP to hide its contents behind *pseudonyms* (PYNs) that are freely-readable by the ISP and that, practically, act as proxies of the actual contents' identities. In fact, the ISP, by counting the occurrences of the pseudonyms, can obtain information about contents' popularity in a privacy-preserving manner. In Section 2.2.2 we briefly review the architecture. For a more in-depth understanding, we refer the reader to [116].

The architecture enables the effective caching of encrypted contents while guaranteeing the protection of users' confidentiality (as contents' popularity can be obtained without decrypting them). However, the architecture does not offer strong protection of contents' popularity privacy. Specifically, the authors of [116] emphasize that the privacy of contents' popularity is at risk, because it can still be inferred from the occurrences of the corresponding PYNs. This holds true mostly for the highly-requested PYNs, which are likely to hide the most popular contents. In addition, an ISP provided with additional information about contents' popularity (e.g., publicly-available hit-parades) can compare the occurrences of the PYNs with the expected popularity of contents. From this comparison, the ISP can guess, with non negligible probability of success, which content is associated with the PYN, thus violating also users' confidentiality.

To handle this issue, the authors of [116] suggest that CPs should frequently change the pseudonym associated with each content, event that we refer to as *refresh*. We observe that a refresh nullifies the knowledge that the ISP was able to acquire about contents' popularity. Hence, increasing the number of refreshes reduces the ability of the ISP to infer valuable information about contents' popularity, which, in turn, improves privacy at the expenses of caching effectiveness. The main contributions of this Chapter are summarized in the following:

- we provide a mathematical formulation of CP's privacy requirements that suits the considered case
- we mathematically formulate a refresh strategy that CPs can employ to improve their privacy

- we perform extensive simulations to quantitatively evaluate the trade-off between privacy and caching effectiveness given by the use of the proposed refresh strategy. Caching effectiveness is measured considering the hit-rate, the retrieval latency and the average traffic load on the network of the ISP

2.1.1 Related Work

An effective cooperation between CPs and ISPs requires the two parties to exchange sensitive or business-specific information and can therefore raise privacy concerns. In this Section, we identify four main types of such information and review existing proposals to allow a cooperation that does not require their disclosure.

The first type of information that ISPs want to protect is that related to their network infrastructures (e.g., network topology). For example, a solution that allows CPs to manage virtual network functions hosted in the ISP domain without gathering information on the ISP's network infrastructure is proposed in [60] and analyzed following a game theoretic approach that proved to be beneficial for both of them. The ISP retains privacy by presenting to the CP only a restricted view of its network infrastructure (e.g., an abstraction of the network).

The second information is the precise users' location, that is owned by the ISP only. The CP, or a Content Delivery Network (CDN) acting on behalf of it, may mis-locate the users and perform the delivery from caches that are far from them. The ISP is aware of the position of its users, and a possible solution consists in allowing the ISP to select the caches from which users are served according to its own objectives, such as to achieve an optimized user experience [52] or an efficient traffic engineering [51]. In [4], we proposed an alternative approach based on a privacy-preserving protocol that makes the ISP able to compute the aggregated number of video requests issued from a particular geographic area and, based on that, cache the videos. Chapter 5 is devoted to the description of this proposed approach.

The third type of information are the identities of the contents requested by users (e.g., their names), that the CP does not want to disclose to the ISP. Specifically, we have identified two main approaches that guarantee the protection of users' requests: i) in the first approach, the CP hides its contents' identities behind pseudonyms that the ISP can analyze to infer useful information about contents popularity without violating users' privacy. Ref. [74] mathematically formalizes the trade-off between privacy and caching resulting from the use of pseudonym-based approaches. Following this approach, the authors of [116] propose a novel cooperative architecture that allows an ISP to cache contents owned by a CP while keeping them in their encrypted form. In this Chapter, we aim to solve the privacy issues that are left unexplored in [116]; in particular, we notice that this architecture is still vulnerable to privacy attacks aimed at obtaining the information about CP's contents popularity. We formally define a data perturbation strategy that the CP can implement to improve privacy, and we quantitatively assess the deterioration of the metrics of caching performance

(e.g., the hit-rate) that this perturbation causes; ii) the second approach focuses on the design of systems that can serve the contents without being able to associate them with the requesting users' identities. For example, the authors of [44] propose a protocol that allows a CDN to serve its users through a peer-to-peer anonymizing network. A similar objective is achieved in [35], which applies a content encryption strategy that allows a CDN to serve contents without identifying them.

The fourth type of salient information that we identified is contents' popularity. In this respect, we proposed in [5] a protocol which permits an ISP and several CPs to cooperate toward the efficient application of caching strategies in a way that is both privacy-preserving and compliant with network neutrality requirements. We devote Chapter 4 to describe this protocol.

2.2 The pseudonym-based approach for a privacy-preserving caching

2.2.1 Content Delivery Scenario

We consider a content deliver system in which the CP owns and maintains a catalogue of video contents that are served to users located within the area of an ISP. The CP is abstracted as a video server that stores the whole video catalogue and is assumed to be external to the area of the ISP. The ISP owns a network infrastructure and provides Internet connectivity to users within a given geographical area. ISP's network nodes may be equipped with caching capabilities that can be used to store and serve (a portion of) the contents owned by the CP.

The content delivery process works as follows: a set of requests $\mathcal{R} = \{r_1, ..., r_N\}$ is issued from the area of the ISP towards the external CP. Each request is assumed to be encrypted by the CP, and the ISP cannot therefore decode the identifier (e.g., the name) of the requested content. We assume that encryption is non-deterministic (e.g., HTTP over TLS protocol [39]), which makes the ISP also unable to guess with non-negligible probability if two requests refer to the same requested content (and therefore also unable to assess the popularity of the content). In the next Section, we briefly review the architecture [116] that allows an ISP to perform effective caching of encrypted contents.

2.2.2 The reference architecture

The authors of [116] introduce a novel functional block in the aforementioned video delivery system, i.e., the *Request Handler* (RH). This element contains information relative to the CP's contents (e.g., the PYNs) and it is designed to be located within the area of the ISP. More in detail, when a new request is issued by the final user, the CP generates a secure token that is delivered to the RH, which utilizes it to extract the PYN corresponding to the requested content. The look-up over the RH database is done under the framework of searchable symmetric encryption (SSE) [36], which allows the contents to be kept in their encrypted version. Once the correct PYN has been

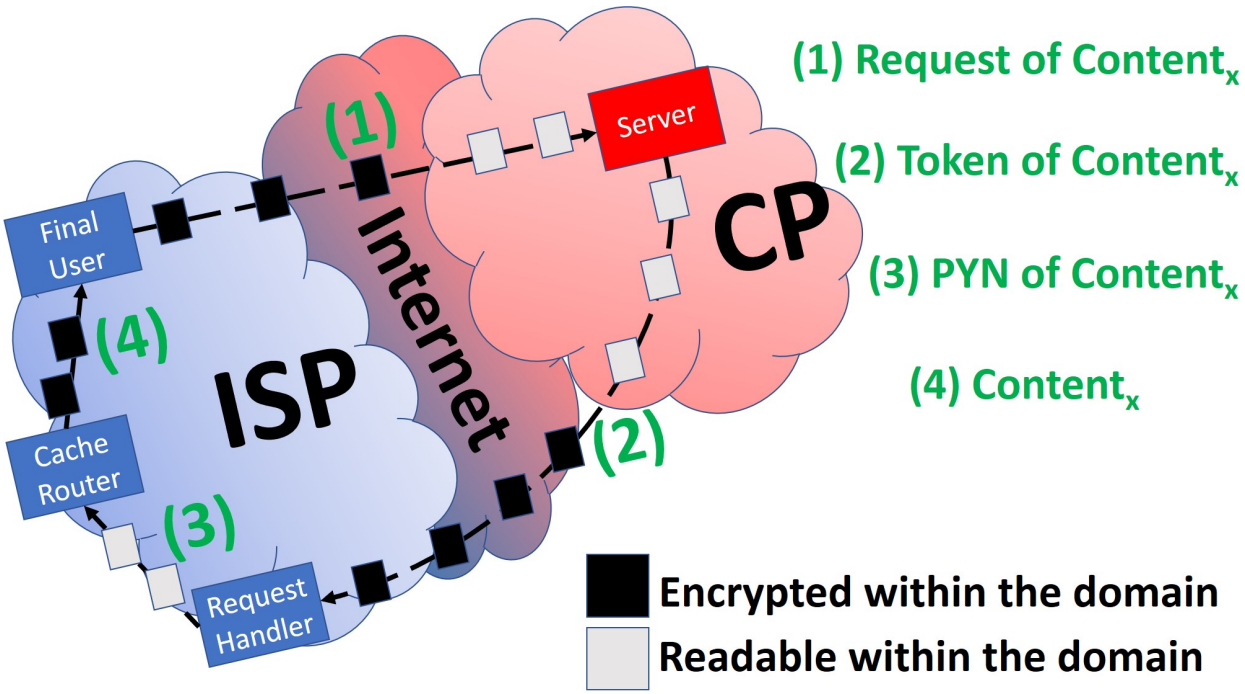


FIGURE 2.1: Representation of the architecture

extracted, the RH uses its up-to-date map to locate the cache server where that content is stored. If the content is available, a redirection procedure toward the CP is issued to directly serve the users from inside the ISP network. The whole request process enabled by the architecture is depicted in Fig. 2.1. For further details, we refer the reader to [116]. Although the proposed architecture is specifically aimed at enabling an effective caching of encrypted contents, several privacy issues remain not totally addressed. We describe them in the following subsection.

2.2.2.1 Privacy Issues

In [116], authors comment about the vulnerability of the proposed architecture to privacy attacks. In particular, they argue that contents' popularity can be leaked by performing statistical analysis on the occurrences of the PYNs. Note, in fact, that after a high number of requests (ideally, an infinite number of requests) the popularity patterns of the PYNs faithfully reflect those of the original contents. Notice also that, in a real scenario (i.e., in which contents are requested a finite number of times) this attack is likely to be more effective on the most popular contents, since their high popularity is evident after a short number of requests. They also mention the possibility that an attacker provided with additional information about content' popularity (e.g., public hit-parades) can even de-anonymize the PYNs by comparing their occurrences with the expected ones (thus violating users' confidentiality). As suggested in [116], the CP can improve its privacy by setting a

lifespan to the PYNs associated with the contents. By renewing the PYNs, in fact, the CP nullifies the knowledge that the ISP had acquired about the popularity of a content.

In the next Section, we more formally describe the trade-off between caching and privacy. To this end, we formally define the concept of contents' popularity, and we propose a definition of privacy suitable for the considered case. Then, we also formalize the refresh-based strategy that CPs can implement to increase their privacy and a countermeasure that the ISP can apply to extract from the PYNs as much knowledge as possible about contents' popularity.

2.3 Trade-off between caching and privacy

The effectiveness of a caching system is generally evaluated considering the Hit-Rate, i.e, the percentage of requests directly served from the cache servers. An improvement of the Hit-Rate corresponds to an increased number of contents that the ISP can serve from sources closer to the end users (i.e., the caches) and, consequently, leads to a reduction of both the average load on the links of the ISP network and of the retrieval latency. In this work we consider three main caching objectives: the maximization of the **hit-rate**, the minimization of the **retrieval latency** and of the **traffic load** on the ISP network links.

The effectiveness of the caching process highly depends on the ability of the ISP to reconstruct the true popularity patterns of the CP's contents from the analysis of the PYNs. We consider all the aforementioned caching objective to be common to both the CP and the ISP. In fact, the CP benefits from delivering its contents over a reliable telecom infrastructure (e.g., characterized by a low congestion probability), while the ISP's reputation increases if the average retrieval latency experienced by its users is reduced [7]. Hence, both the ISP and the CP are incentivized that the reconstruction of the popularity patterns is as much faithful as possible. However, the CP is also concerned that this process does not violate its privacy, which is formally defined in the next subsection.

2.3.1 Privacy of Contents' Popularity

We refer to n_i to indicate the number of requests that, during a considered period, a CP receives for the i-th content of its catalogue $\mathcal{C} = \{C_1, C_2, ..., C_M\}$. We define the popularity of the i-th content as its position on the rank of the number of requests, sorted in descending order. Note that the most requested content has popularity equal to 1, while the least requested content has rank M. As commonly assumed in the literature, contents' popularity distribution is modeled with the Zipf function, that effectively captures its long-tail effects (the number of highly-popular contents is a small percentage of the total number of contents).

As previously stated, the architecture proposed in [116] exposes the occurrences of the PYNs to the ISP, from which it is possible to obtain valuable insights about CP's popularity patterns (e.g., to

measure the popularity gap between most and average popular contents). We propose a definition of privacy that measures the information about contents' popularity that can be extracted from the analysis of the PYNs. Specifically, we define the privacy of the t-th popular content as follows:

$$Privacy(C^{(t)}) = 1 - P(C^{(\tau)}|PYN^{(t)}), \quad \tau \in \{t-\eta, t+\eta\} \tag{2.1}$$

where $C^{(t)}, C^{(\tau)}$ are the t-th and τ-th most popular contents, respectively, and $PYN^{(t)}$ is the t-th popular PYN. $P(C^{(\tau)}|PYN^{(t)})$ is the probability that the t-th most popular PYN hides the τ-th most popular content, where $\tau \in [-\eta + t, t + \eta]$. In other words, this probability measures the information obtainable about the popularity of a content given the popularity of the corresponding PYN. η defines the tolerance that the CP has in considering its privacy violated. For example, a high η implies that the CP is not even willing to expose coarse information about contents' popularity (e.g., the ISP should not be able to know if a PYN refers to a highly-popular or to an average popular content). Instead, by assuming a small η, the CP considers its privacy violated only if the ISP can accurately infer the popularity of a content from that of the corresponding PYN.

2.3.2 Protection and Attack to Contents' Popularity Privacy

2.3.2.1 Security Model

When using the architecture, the ISP and the CP do not expose the same amount of information to the other party. Specifically, the ISP is not required to reveal to the CP any data, while the CP exposes the PYNs. Hence, the two parties are not equally susceptible to privacy attacks.

Accordingly, we model the CP as a *honest* entity that does not deviate from the operations required for an effective use of the architecture. As for the ISP, we model it as a *honest-but-curious* entity that honestly executes the protocol prescribed by the architecture, but also tries to infer as much knowledge as possible from its transcripts (i.e., the PYNs and their number of occurrences). We assume that the ISP is economically-incentivized to infer contents' popularity from the analysis of the PYNs mainly to improve the efficiency of caching (i.e., to store the contents that are actually the most popular ones). Another reason to model the ISP as honest-but-curious is that, by tracking the PYNs requested by users, the ISP might also able to obtain salient and business-critical information, i) such as the popularity patterns of the CP's catalogue, ii) the similarity (resp., dissimilarity) among users who often (resp., rarely) request the same PYNs and iii) if eventually provided with additional information (e.g., public hit-parades), also their preferences.

Hence, the CP is willing to keep the information that the ISP can extract from the PYNs below a given threshold. To this end, as suggested in [116], it can frequently renew the association between contents' identifiers and PYNs. In the next subsections, we formally define a PYNs' renewal approach that is in line with the suggestions of [116] and we propose a countermeasure that the ISP can apply to extract as much knowledge as possible from the analysis of the PYNs.

2.3.3 PYNs' renewal strategy

We refer to the event of setting a new PYN for a content to as *refresh*, and we formalize a *refresh-based* renewal strategy in which the PYNs associated to all the contents of the calagoue are refreshed simultaneously. Formally, given a period T in which the CP performs R refreshes, the τ-th popular content $C^{(\tau)}$ is associated, during its lifespan within the ISP network (i.e., $0 < t \leq T$), to R different PYNs. During the r-th refresh period, $C^{(\tau)}$ is associated with a PYN that, within this period, has popularity rank j_r. In formulas,

$$PYN(C_\tau) = \begin{cases} PYN_1^{(j_1)}, & 0 \leq t \leq \frac{T}{R} \\ \cdots & \\ PYN_R^{(j_R)}, & \frac{R-1}{R} \cdot T < t \leq T \end{cases} \tag{2.2}$$

where C_τ is the τ-th popular content in the period $0 < t \leq T$ and $PYN_r^{(j_r)}$ is the j_r-th popular PYN within the r-th refresh period. By increasing the number of refresh R, the CP improves privacy at the expenses of caching, since it nullifies the knowledge that the ISP was able to obtain until that moment about contents' popularity. In the next subsection, we describe a countermeasure that the ISP can apply to recover useful information about contents' popularity.

2.3.4 Attack to the privacy

Starting from the intuition that the occurrences of the requests for the contents and for the corresponding PYNs are likely to share some statistical characteristics (e.g., number, interarrival times, etc...), we propose a heuristic strategy to rank the PYNs in such a way that their order within each refresh round likely reflects the popularity rank of the contents hidden behind them. PYNs with the same popularity rank in different refresh rounds are then clustered together; the PYNs within the same cluster are considered the ones that have most likely been associated with a given content. The heuristic strategy is described in detail in the following:

1. The ISP identifies the number of refresh periods. Notice that the transition from a refresh period to the successive one is characterized by a sudden termination of the requests for the currently-employed PYNs and is therefore easy to identify.

2. Then, the ISP computes a value $\theta_r^z = \frac{\mu_r^z}{N_r^z}, \forall(z, r)$, where z and r are the indexes of a generic PYN and of the considered refresh period, respectively. μ^z and N^z are the average interarrival times and the number of occurrences of the z-th PYN.

3. In the third phase, the ISP sorts θ_r^z in ascending order, in such a way that the first PYNs (i.e., characterized by the lowest average interarrival periods and by the highest number of occurrences) are likely associated with the most popular contents.

4. Finally, the ISP considers the R PYNs at the j-t position to be associated with the j-th most popular content.

Assuming that the ISP never gets in possession of the actual association between PYNs and their relative contents, the only way that it has to verify the success of its attack is by observing an improvement of caching performance. On the other hand, the CP has the knowledge of both the occurrences of the PYNs and the actual occurrences of the contents and, by implementing a similar attack strategy, it can use Eq. 2.1 to measure the privacy of the τ-th popular content as $\frac{N_{PYN}^{(\tau)}}{R}$ considering that $N_{PYN}^{(\tau)}$ is the number of PYNs associated with the τ-th popular contents whose PYN has popularity t, whereas R is the number of PYNs whose popularity rank is t. On the other hand, the CP has the knowledge of both the occurrences of the PYNs and the actual occurrences of the contents and it can measure the privacy of the t-th as follows:

$$Privacy(C^{(t)}) = \frac{N_{PYN}^{(\tau)}}{R} \tag{2.3}$$

Notice that Eq. 2.3 is derived from Eq. 2.1 considering that $N_{PYN}^{(\tau)}$ is the number of PYNs at the t-th position of the rank (computed using the aforementioned attack strategy) that hide a content whose popularity is $\tau \in \{t - \eta, t + \eta\}$; R is the number of refresh rounds, which is equal to the number of PYNs associated with a content during its lifespan.

2.4 Numerical Results

2.4.1 Simulation Settings

We perform simulations to i) compute the probability that privacy is violated and ii) to assess the performance of the caching system. The former objective is achieved by empirically computing the probability according to Eq. (2.3); as far as the latter is concerned, we employ a simulator [104] designed to study the performance of caching systems within an Information Centric Network (ICN) context [1]. ICN was proposed to make content delivery in Internet more scalable with respect to the host-centric paradigm and the architecture presented in [116] is easily applicable to our scenario. In practice, instead of searching the contents by their actual identities, users (i.e., the *receivers*) first obtain the corresponding PYN from the Request Handler, and then perform a look up of the nodes (i.e., the *sources*) that have cached it. The employed methodology to assess the trade-off between privacy and caching is applicable also to the host-centric paradigm, while a specific research is needed to assess the generalization of the obtained results as well.

2.4.1.1 Caching Strategies

We employ two caching strategies, namely the *Cache Less For More* (CL4M) [29] and the *Leave Copy Down* (LCD) [72]. The former caches a content in the node belonging the highest number of

paths between a source and a receiver. The latter caches a content in the cache immediately after the source, in the direction of the receiver. We employ the *Least Recently Used* (LRU) as caching eviction policy.

2.4.1.2 ISPs Network

We consider the topology of the backbone TISCALI network described in [111] that is composed of 240 nodes and 404 links connecting them. According to the implementation of the simulator, each link introduces a delay of 2 ms and its capacity is assumed to be sufficient to accommodate all the requests. 36 among the available nodes are randomly selected to serve as cache servers and are assumed to host, in total, a maximum of 2500 contents.

2.4.1.3 Description of the requests and of the CP's catalogues

We employ a dataset of real VoD requests that covers a period of 6 months (from October 2011 to March 2012) and has been collected at the Lancaster University, where a Laboratory provides a small IPTV service available for both the campus university and a small village nearby. The dataset is presented and described in detail in [47]. We parsed the dataset in order to discard irrelevant data (e.g., contents transmitted over radio channels) and we obtained a reduced dataset of around $90K$ requests of 3100 different video contents, either live or on-demand. This dataset allows us to conduct realistic researches, mainly because content popularity evolution is not a easy-to-model phenomenon.

However, the domain from which it has been collected (i.e., University Laboratory/Campus) is not general enough for our purposes (i.e., cooperation between large-scale autonomous systems). Therefore, we decided to craft also a synthetic dataset that enables to target our research toward its core objective. We employ four different traces, characterized by the combination of two parameters, namely the number of contents $N_c \in \{5000, 10000\}$ and the skew value $\alpha \in \{0.7, 0.9\}$ of the Zipf distribution [24], which is widely used to model contents' popularity. We assume that the requests for the synthetic catalogue are issued at a rate of 1 req/sec for a total period of 1 day. At this rate, the total number of request within one day is quite similar to the number of requests of the real dataset (i.e., $90K$). Hence, we assume that the requests for the latter one are issued within a 1-day period as well. We set the size of each content to $1.5Gb$.

2.4.2 Discussion

In this Section, we show the average of the results obtained by performing 500 simulations with varying number of refreshes R, contents' catalogue characteristics, caching strategy and privacy threshold η. In Fig. 2.2 we show how privacy improves at the expenses of a degradation of the Hit-Ratio obtained employing the CL4M caching strategy over a synthetic traffic trace (with $\alpha = 0.9$ and $N_c = 5000$). Privacy is measured as the probability that the most popular PYN (within a

Table 2.1: Hit-Rate (%)

		Caching Strategy						
Catalogue		NC	CL4M			LCD		
α	$\mathbf{N_c}$	$\forall$**R**	**1R**	**5R**	**10R**	**1R**	**5R**	**10R**
0.7	$10K$	0	25.6	21.4	19.1	32.1	30.4	29.4
0.7	$5K$	0	33.9	30	25.9	42.5	41.2	39.6
0.9	$10K$	0	45.8	41.4	38.6	51.6	50	48.7
0.9	$5K$	0	52.1	47.7	44.9	59.5	57.8	56.2
Lancaster		0	76.7	73.8	71.9	84.6	83	81.3

Table 2.2: Average Retrieval Latency (ms)

		Caching Strategy						
Catalogue		NC	CL4M			LCD		
α	$\mathbf{N_c}$	$\forall$**R**	**1R**	**5R**	**10R**	**1R**	**5R**	**10R**
0.7	$10K$	92.1	72.9	75.9	77.7	68.2	69.5	70.4
0.7	$5K$	92.1	66.6	69.5	72.5	60.6	61.6	62.9
0.9	$10K$	92.1	56.7	60.1	62.2	52.5	53.8	54.9
0.9	$5K$	92.1	51.9	55.2	57.4	46.6	47.9	49.3
Lancaster		92.1	32	34.17	35.7	26.7	28	29.5

Table 2.3: Average Load on ISP's network links (Gb)

		Caching Strategy						
Catalogue		NC	CL4M			LCD		
α	$\mathbf{N_c}$	$\forall$**R**	**1R**	**5R**	**10R**	**1R**	**5R**	**10R**
0.7	$10K$	28.6	26.6	26.8	27	26.3	26.5	26.7
0.7	$5K$	28.6	25.9	26.2	26.3	25.6	25.9	26.1
0.9	$10K$	28.6	23.8	24.1	24.4	23.4	23.6	23.9
0.9	$5K$	28.6	23.1	23.5	23.8	22.7	22.9	23.3
Lancaster		28.6	19.2	19.5	19.9	19.4	19.5	20

given refresh period) is not associated with the most popular content, considering different privacy thresholds $\eta \in \{0, 1, 2, 3\}$.

Figure 2.2 shows that the Hit-Ratio decreases steadily with increasing R as it drops from a maximum of 52% (for $R = 1$) to a minimum of 44% (for $R = 10$). We also observe that privacy improves more significantly with increasing R, i.e., for $R > 3$. This shows that the CP can tune the value of R to improve the privacy while limiting the degradation of caching performance. Moreover,

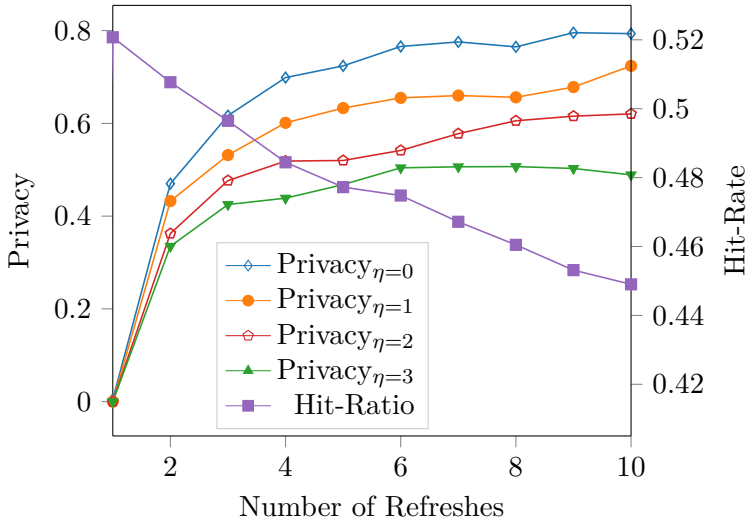
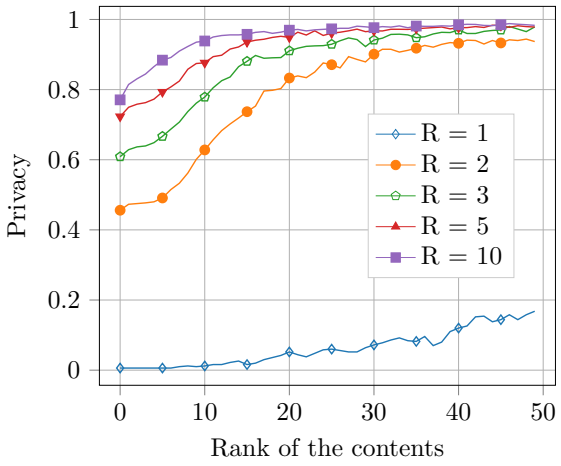


Figure 2.2: Trade-off between Caching Hit-Ratio and Privacy for different values of the privacy threshold η computed for a catalogue of $N_c = 5K$ contents and skew parameter $\alpha = 0.9$

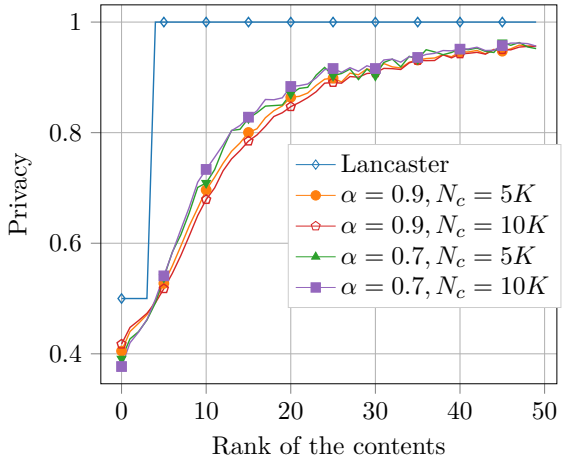
results show, as expected, that the value of the privacy threshold η plays an important role. In fact, we notice that the privacy of the most popular content increases from 0.48 to 0.79 when η goes from 0 to 3 (i.e., when privacy is considered violated if the most popular PYN is associated with the most popular contents or, instead, with one of the 4 most popular contents).

We then extend the analysis of the degradation of caching to all the available traffic traces (4 synthetic and one real) using the No caching (NC), the Cache Less For More (CL4M) and the Leave Copy Down (LCD) caching strategies. The results for the Hit-Ratio, the average retrieval latency and the average load on the ISP's network links, obtained for $R \in \{1, 5, 10\}$ are summarized in Tab. 2.1, Tab. 2.2 and Tab. 2.3, respectively. Results show that all three metrics degrade for increasing R. For example, the Hit-Ratio for Lancaster content catalogue decreases from 52.1% to 44.9% and from 59.5% to 56.2% for the CL4M and LCD caching strategies, respectively. As for the average load on ISP's network links and the average retrieval latency, they both slightly increase (by an average of 3% and 10% respectively) for an increasing R (from $R = 1$ to $R = 10$) for both the considered caching strategies. However, it is important to highlight that the performance of the caching strategies are different. This shows that different caching strategies present different degree of robustness toward the use of the architecture [116], and, in particular, toward the refresh-based anonymization approach.

Now, we evaluate the impact of the number of refreshes and of the characteristics of the catalogue (e.g., number of contents) on the privacy of the 50 most popular contents. We set the privacy threshold $\eta = 0$. Fig. 2.3a shows the privacy computed for the synthetic traffic trace with $\alpha = 0.9$ and $N_c = 10000$ for several numbers of refreshes ($R \in \{1, 2, 3, 5, 10\}$). Results confirm that the incremental improvement of privacy is higher for lower values of R and this is particularly evident



(a) Impact of R on privacy considering a catalogue of $N_c = 10K$ contents and skew parameter $\alpha = 0.9$



(b) Impact of the catalogue's characteristics on privacy considering $R = 2$ refreshes

FIGURE 2.3: Privacy of the most popular contents of the CP's catalogue considering a privacy threshold $\eta = 0$

for the least popular contents. For instance, the 50-th popular content experiences an improvement of privacy of 0.7 (compared to an improvement of 0.43 for the most popular content) when R goes from 1 to 2.

Finally, in Fig. 2.3b we evaluate the privacy for all the catalogues for R =2. We notice that the privacy computed for the real dataset is significantly higher than the privacy computed for the synthetic ones. This is due to the fact that the popularity of the most popular contents in the Lancaster dataset are much more similar than what is observed in a Zipf-distributed catalogue, and this makes it harder to properly assess their popularity ranks by analyzing the occurrences of the PYNs. On the other hand, we also notice that the skew parameter α affects privacy more than the number of contents N_c, whose influence can be considered negligible.

2.5 Concluding Remarks

2.5.1 Summary

In this Chapter, we considered a content delivery system composed of a Content Provider (CP) and a Internet Service Provider (ISP) that cooperate by using an emerging architecture aimed at allowing the ISP to cache the CP's contents without violating its privacy requirements, e.g., the information about its contents' popularity. The architecture enables the CP to hide its contents behind pseudonyms that are visible to the ISP and this threatens its privacy. We formalized a strategy that the CP can follow to improve privacy, and we evaluated the resulting trade-off with caching, measured considering the hit-rate, the retrieval latency and the average load on the links of

the ISP network. The results, obtained by performing simulations over both real and synthetic data, show that a significant improvement of privacy can be achieved at reduced costs in terms of caching performance degradation and suggest the applicability of the architecture in a real scenario.

2.5.2 Final Comments

In this Chapter, we have seen that, when encryption strategies are applied, effective caching can only be performed if ISPs and CPs implement a form of cooperation. As the use of encryption is steadily increasing, we should also expect an increasing number of such cooperative schemes. In this context, since the available caching storage of an ISP is limited, potentially-discriminatory issues may arise when multiple CPs are willing to exploit it to better serve their users. In the next Chapter, we elaborate on the principles that make such forms of cooperation non discriminatory towards the CPs, i.e., compliant with Network Neutrality ideals.

Network-Neutrality Compliant Caching 3

As shown in the previous Chapter, caching is an important tool to reduce ISPs' network traffic and to increase customers' QoE. Let us notice that, since the available cache storage is limited, the implementation of caching strategies requires to chooce the portion of contents served from the caches. Therefore, caching is an inherently-selective process that results in service differentiation and, potentially, discriminatory treatment of the CPs that exploit the caching system. Despite of this, in-network caching is not generally regarded as a traffic differentiation technique and, therefore, is not considered by Network Neutrality (NN) regulations. In this Chapter, we look at this problem from a different perspective: in the today's Internet, ISPs are prevented from applying in-network caching without cooperating with CPs. For instance, the wide use of encryption makes the ISPs unable to identify the contents worth caching (i.e., the most requested ones). In this scenario, where the cooperation between ISPs and CPs seems mandatory, a rigorous definition of NN-compliant caching is needed to agree on the principles that make caching fair towards users and CPs. To this end, in this Chapter we evaluate the QoE that CPs are able to offer to their users under different models of cooperation with an ISP. We observe that the QoE is significantly affected by the adopted cooperative model, and we conclude that caching may lead to discrimination. Finally, we suggest a few research directions towards the implementation of non-discriminatory (i.e., NN-neutral) in-network caching.

3.1 Motivation

The Internet is constantly evolving to adapt to the new services offered by CPs and to meet the increasing expectations of their users (e.g., in terms of QoE). For example, ISPs implement caching strategies that CPs employ to serve a portion of their contents directly from the cache servers, thus

increasing the QoE experienced by their users. Since multiple CPs aim at exploiting the caching system of the ISP, caching may lead to discriminatory issues that should be carefully analized in a context of Network-Neutral Internet. Our aim in this Chapter is to answer to the following pending question: how can caching be performed to be both efficient and non discriminatory towards the CPs (i.e., *NN-Compliant* caching)? Current NN definitions are built on the principle that traffic can differentiated based on its QoS requirements (e.g., VoD can be prioritized over e-mail exchange), but they do not provide rigorous guidelines to differentiate contents of the same class (VoD, in the considered case). Moerover, today's Internet is characterized by several properties that, in our view, should be considered in the debate about NN-compliant caching:

- Encryption: CPs are increasingly encrypting the traffic destined to their users to ensure privacy and security. For instance, as of year 2019, 90% of the most accessed web sites in Internet use the HTTPS protocol as default option[1]. This fact prevents ISPs from applying even basic forms of traffic differentiation. For example, how can an ISP decide which contents to cache without being able to inspect them? In fact, requests for the same content are indistinguishable under encryption schemes, and this does not allow the ISP to infer the content popularity. A possible solution to this issue is to make CPs and ISPs jointly manage the caches (e.g., by employing the architecture described in the previous Chapter to enable the effective caching of encrypted contents). This solution, however, calls for cooperation schemes that inevitably open the doors for possible violations of the NN principles. For example, the arbitrary selection of the counterparts of the cooperation (i.e., the CPs that the ISP decides to cooperate with) has to be carefully regulated.

- Cooperative Strategies: nowadays ISPs commonly employ Virtualization to easily instantiate resources and offer specific services, thus widening the spectrum of the potential cooperation schemes that ISPs and CPs can implement. With ISPs that become more than simple providers of connectivity, however, new concerns are raised about their discriminatory behaviour.

To shed some light on the NN issues arising from the application of caching strategies, in this Chapter we consider three different cooperative approaches that ISPs and CPs can implement. These approaches are referred to as i) *partially-cooperative*, ii) *non-cooperative* and iii) *fully-cooperative*. In the first one, only a subset of all the CPs cooperate with the ISP to realize caching strategies within its network. This approach is discriminatory towards the CPs that do not cooperate. On the other hand, the second and the third approaches can be considered NN-compliant at different degrees. Specifically, in the second one all the CPs manage the same amount of caching resources inside the network of the ISP, while in the third one each CP manages a portion of cache storage proportional to the popularity of its contents.

[1] https://transparencyreport.google.com/https/overview

We provide a numerical assessment of the effectiveness of these approaches, which are evaluated considering the Hit-Ratio experienced by the ISP and by the involved CPs. From this evaluation, we conclude that the employed cooperative approach may lead to a discriminaton of the involved CPs. From this assessment, we then provide a possible definition of NN-compliant caching. Specifically, we consider the fully-cooperative approach to be the one most in line with NN requirements, since it leads to the highest Hit-Ratio experienced by the ISP while discriminating the CPs based on an objective criterion, i.e., their popularity, and not based on other forms of arbitrary agreement. Finally, we envision the implementation of an open protocol to enable the fully-cooperative approach while guaranteeing privacy to CPs and ISP. In particular, our aim is to develop a protocol to apply NN-caching in presence of encrypted contents. In the last part of the Chapter, we review existing approaches to perform a privacy-preserving caching, and we show how they provide guidelines to realize such protocol. We propose an implementation of this protocol, and we describe it in the next Chapter.

3.2 Towards a Net-Neutrality Definition for in-network Caching

3.2.1 The Partially-Cooperative Caching

Nowadays, the highest portion of traffic inside ISPs' networks is generated by the delivery of VoD contents owned by few giant CPs. For instance, the contents streamed by Netflix users account for an impressive percentage of the global Internet traffic (up to 40% at peak hours in the USA) [95]. It is common that these big CPs deploy cache servers inside the ISPs domains to improve the streaming experience of the users and, at the same time, significantly reduce network traffic. This is a win-win solution that apparently satisfies all the involved entities: the contracting CP, the users, ISPs and, to some extent, also competitor CPs, which can count on a less congested network to deliver their services. However, this solution can also be considered discriminatory, because contents are treated differently based on the CP ownership. There is no clear cut on whether this type of cooperation should be considered a case of NN infringement. Following the former argument, some believe it is not. ISPs are behaving neutrally, as they seek their advantage at no expense of other players and without negatively affecting the Internet community. Conversely, some others see this strategy as a subtle form of traffic prioritization, which should therefore be prohibited in a regime of content-based indiscrimination.

We argue that the partially-cooperative strategy would not violate NN provided that a similar solution could be applied by every CP. However, such condition turns out to be difficult to achieve in practice. As an example, let us consider the case of two CPs operating in the same sector (e.g., VoD), but with significantly-different market power. The smaller CP is clearly disadvantaged as it does not have scale to negotiate this kind of agreement with the ISPs. The result therefore is that its users perceive a lower QoE, which confirms that this type of cooperative approach can lead to

discrimination.

Being a vital instrument for the effective management of ISPs' resources, however, cooperative caching cannot simply be prohibited in favor of a pure non-discriminatory principle. In our view, ISPs are perfectly legitimate to apply an effective caching strategy, as long as this is done avoiding any discrimination among the various CPs. In the next subsection, we describe more in detail the idea of NN-compliant caching considered in this Chapter, and we present two approaches to realize it, namely the non-cooperative and the fully-cooperative one.

3.2.2 NN-compliant caching

A possible vision of NN-compliant caching is based on the idea that the cooperation between ISPs and CPs must avoid arbitrary forms of agreements and should, instead, be done according to some objective criteria. A simple criterion consists in assigning the same cache storage to all the CPs that are willing to exploit the caching system of the ISP. This is the approach referred to as non-cooperative, as it does not require ISP and CPs to implement any scheme of cooperation to establish the amount of cache storage that each CP is entitled to manage. This approach, however, is also highly-inefficient, as it discards the different level of attractiveness of the CPs. Indeed, content popularity is the main factor leading to good caching performance (e.g., high hit-rates).

Hence, following reference [84], we consider contents' popularity to be an objective criterion according to which it is possible to implement caching strategies that are both efficient (as the most requested contents are cached) and neutral towards the CPs. In fact, while being aware that this approach clearly favors big CPs (as they are expected to own most of the popular content), we also believe it is non-discriminatory because it favours a CP based on its ability to engage the final users. In this way, the network is neutral with respect to both the CPs and the users, which are not driven to request a content because of a strategic cooperation. Instead, contents are awarded in terms of superior QoS, which results in superior QoE only by virtue of their popularity.

To avoid potentially-discriminatory solutions, ISPs may employ transparent caching technologies [64] to inspect the traffic traversing their network and infer content popularity. If encryption schemes are in place, however, such techniques are ineffective and alternative strategies to manage the caches are required to avoid using single-purpose caching systems like those envisioned in the partially-cooperative caching paradigm. In the next Section, we provide a quantitative assessment of the traffic distortion induced by the three considered cooperative models, which are compliant with NN at different degrees.

3.3 Quantitative comparison of caching frameworks

The performance of a caching strategy is typically measured considering the Hit-Rate, i.e., the percentage of requests directly served from inside the domain where the caches are located (the ISP,

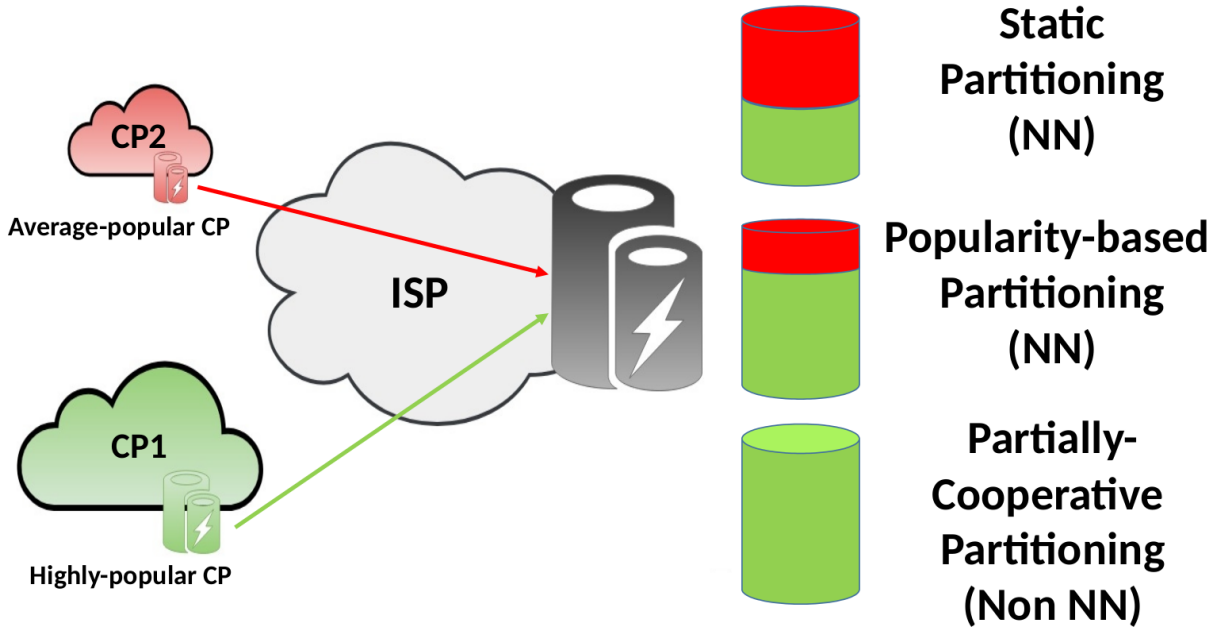


FIGURE 3.1: Representation of the considered scenario and of the employed partitioning strategies.

in our case). Hence, a high Hit-Rate is always desirable as it leads to a significant reduction of the traffic volume across the network.

We now consider three different scenarios of relations between two CPs and an ISP. As depicted in Fig. 3.1, the CPs compete to obtain a portion of the cache storage made available by the ISP in order to offer a better service to their users. CP_1 and CP_2 are assumed to have catalogues of different sizes and with different popularity patterns. Specifically, CP_1 is considered a CP with high attractiveness and most of its contents are more popular than those offered by CP_2. We assume that each CP stores its most popular contents, which remain fixed during the considered time period.

In the first scenario, referred to as Static Cache Partitioning, ISP and CP follow the non-cooperative approach: the ISP equally shares the storage of its cache among the CPs, regardless of the global popularity of the contents that they own. This is the most neutral case, and we consider it as the baseline for the performance comparison. In order to allow for content encryption by the CPs, we assume that CPs manage their caching portion autonomously.

In the second scenario, referred to as Popularity-based Cache Partitioning, the ISP selects the contents to cache only according to their global popularity, i.e., regardless of the owner CP. In case the contents are not encrypted, the ISP can infer which contents are the most popular ones. In this way, it is guaranteed that the available cache storage is divided among the involved parties proportionally to the number of the most popular contents available in their catalogues. In this way,

the CP that owns more popular content is assigned a larger slice of cache storage. This approach aims at maximizing the Hit-Rate of the ISP and it is not discriminatory. However, as encryption is nowadays widely employed in the content delivery chain, ISP and CPs need to jointly compute the amount of storage that should be dedicated to each party. In Section 3.3.1, we describe how two existing techniques can provide several guidelines for the development of protocols enabling a seamless and non-discriminatory full-cooperative approach. If these protocols are not employed, however, ISPs are pushed to follow the partially-cooperative approach, i.e., to behave in a non-NN manner. As an example of partial cooperation, we consider a third scenario, referred to as Partially-Cooperative Cache Partitioning, that aims at representing a realistic situation, where the ISP cooperates only with CP_1. Because in this study we consider two CPs only, the whole cache storage is assigned to CP_1.

With the aim of understanding which caching solutions may lead to discrimination, we perform a simulative study assuming that the two CPs offer catalogues of significantly-different degrees of attractiveness. Specifically, we assume that CP_1 and CP_2 own 67% and 33% of the most popular contents, respectively. This proportion can be obtained by properly choosing the sizes of the offered catalogues and the skewness of their contents popularity. Note that infinite combinations of such parameters can lead to the target proportion. In our simulations, we arbitrarily assume that CP_1 and CP_2 offer a catalogue of 50K and 400K contents, respectively, whose popularity follows the Zipf law [24] with skewness parameter $\alpha = 0.8$ and $\alpha = 0.9$, respectively. We recall that the skewness of the content popularity distribution augments with increasing α, i.e., fewer contents are highly-requested with α approaching one. The domain of the ISP is abstracted as a single cache with a total storage of 40K contents.

In Fig. 3.2, we show the Hit-Rates experienced by the two CPs depending on the employed cooperative strategy. The Static Partitioning approach penalizes CP_1 and benefits CP_2, which receives more cache storage than what it would deserve based on the global popularity of its contents. In fact, CP_1 sees a significant increase of its performance if, instead, the Popularity-based Partitioning approach is used. Conversely, CP_2 is penalized by such fair assignment of resources. CP_1 maximally benefits from the Partially-Cooperative Partitioning approach, while CP_2 experiences a null Hit-Rate because none of its contents are cached, which results in a discriminatory treatment. Fig. 3.2 also shows that the ISP significantly benefits from being net-neutral. However, the presented results are not sufficient to state that the ISP would always prefer to employ a NN-compliant caching. In fact, we are aware the maximization of the Hit-Rate is just one of its possible objectives. For example, the ISP may be concerned of performing an effective Traffic Engineering (e.g., to minimize its resource occupation), which depends on other factors beside the global popularity of contents (e.g., their size). In the next Chapter, we provide a more in-depth analysis of the caching performance obtained following a popularity-based caching subdivision, and we present the protocol designed to enforce it in a privacy-preserving manner.

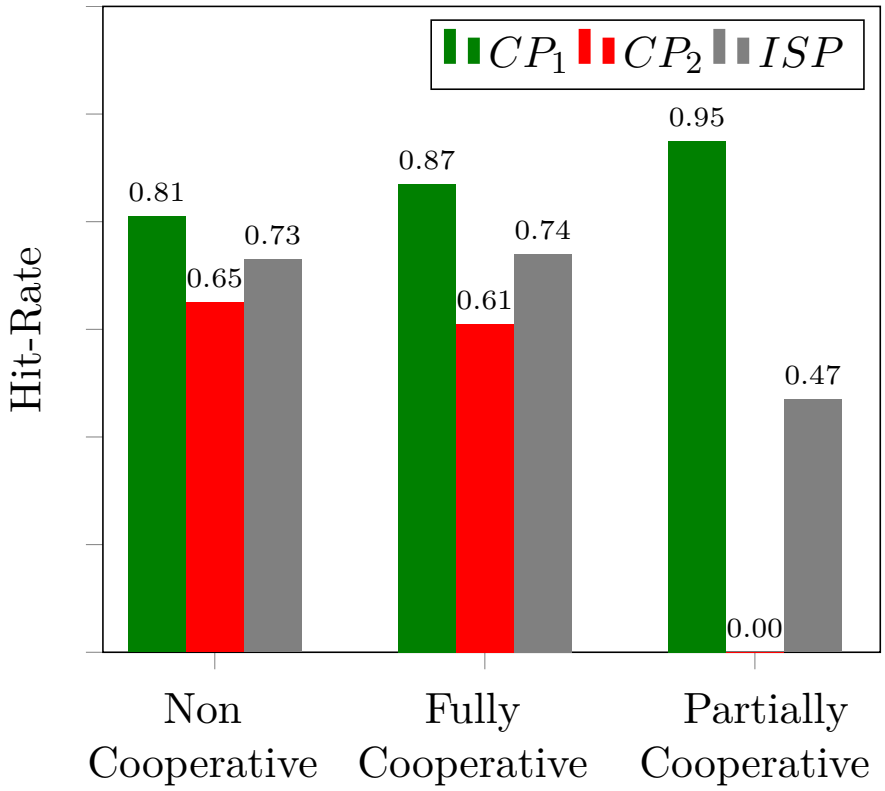


FIGURE 3.2: Hit-Rates measured by CP_1, CP_2 and ISP for different cooperative schemes

3.3.1 Discussion and Future work

Since caching causes a differentiation of traffic generated by the CPs, the topic of in-network caching should be fully integrated in the NN debate. To take a first step in this direction, in this Section we present a possible definition of NN-compliant in-network caching, and we suggest few research directions toward its enforcement.

3.3.1.1 Definition of Network-Neutral in-network caching

Let us define NN-compliant in-network caching as the caching process that causes a traffic differentiation only on the basis of the global popularity of contents. Being aware that the maximization of the Hit-Rate is just one of the goals of the ISP, we also advocate a more general definition of NN-compliant caching, that is able to include other objectives based on which the ISP can legitimately perform traffic differentiation (e.g., minimizing the traffic load over their networks).

3.3.1.2 Open Questions

The fulfilment of the aforementioned principle, however, is practically hindered in a scenario of all-encrypted web, in which ISPs are incapable of inspecting the contents traversing their networks to infer their global popularity. A viable alternative would be to make CPs and ISPs cooperate with the goal of jointly managing the cache systems, which raises the following questions:

- given that ISPs are prevented from identifying the global popularity of contents, how is it possible to implement a cooperative system that does not lead to a discrimination among the CPs?
- given that contents' popularity is a business-critical asset that CPs are not willing to expose (as discussed in the previous Chapter), how can the ISP balance the available cache storage without requiring the CPs to reveal the popularity of their contents?

3.3.1.3 Research Directions

We advocate the definition of an open protocol enabling any potential CP to receive the amount of storage that it is entitled to manage according to the popularity of its contents. In our view, such protocol should meet three main requirements:

- NN-compliance that, in our vision, means to allow the ISP to maximize the Hit-Rate by favouring the CPs only by virtue of the attractiveness of their catalogue, i.e., only because of the global popularity of their contents
- Privacy to allow the CPs not to disclose information about the popularity of their contents, which is a business-critical information
- Scalability to accommodate the demands of a potentially high number of CPs and contents

To the best of our knowledge, existing methods that can be used to develop such protocol follow two different approaches, that we describe in the following from a high-level standpoint. In addition, we analyze the extent to which such approaches meet the three aforementioned requirements. The two approaches treat the contents of the involved CPs differently. However, under the definition of NN-compliant caching discussed throughout this Chapter, both are non-discriminatory.

One of them has been proposed in [116] and considered in the work described in the previous Chapter. This solution envisions a content delivery architecture that allows to efficiently cache, in the domain of the ISP, contents encrypted by a CP. Each request issued to the CP results in the generation of a pseudonym (which univocally identifies the requested content) that is freely readable by the ISP. Based on this pseudonym, the ISP can locate the requested content and directly serve it to the user. By employing this architecture, two main goals are achieved: (i) the ISP can count the occurrences of the pseudonyms to infer the popularity of contents and, based on that, apply

any known caching strategy; (ii) the CPs can keep their contents encrypted to ensure security and privacy to their users without preventing the application of caching.

Assuming that the ISP aims at maximizing the Hit-Rate, each CP receives a fair amount of storage based on the global popularity of its contents. The privacy of the CPs is guaranteed as long as the ISP is not able to infer the actual content names from their pseudonyms. However, as detailed in the previous Chapter, the ISP can obtain many valuable information about contents' popularity from the analysis of the pseudonyms. Moreover, since the information about the most popular contents can be considered public, to some extent (e.g., the hit-parade is typically publicly known), the ISP may be able to associate the highly-popular contents with the most frequently occurring pseudonyms.

To cope with this issue, the association between contents and their pseudonyms can be refreshed with some frequency. Such procedures nullify the acquired knowledge of popularity, therefore improving privacy at the cost of deteriorating the performance of caching. Concerning scalability issues, the authors of [116] state that the proposed solution can easily handle a large amount of requests (up to hundreds of thousand users even for a basic implementation of the architecture). In practice, the architecture can be optimized to be deployed in real scenarios (e.g., by increasing the computational power of its main functional blocks).

In the other approach, which is proposed in [10], the ISP cooperates with several CPs and reserves a portion of its cache storage to each of them. The ISP can obtain the information about the Hit-Rate experienced by each individual CP by virtue of the resource occupation induced by that CP on its network. Then, the ISP executes an iterative algorithm that takes as input the Hit-Rates of the individual CPs and converges to the optimum partitioning (i.e., the partitioning that maximizes the global Hit-Rate measured by the ISP). Note that the individual Hit-Rates are the only information that CPs are required to disclose to the ISP, which makes the approach privacy-preserving. One fundamental difference with respect to the former approach is that the ISP is only in charge of partitioning the available storage, while caching is performed by the individual CPs. Practically, this approach is a way to extend the footprint of the CPs by giving them a proper amount of cache resources inside the domain of the ISP. Scalability is analyzed in detail in [10], where it is shown that the computational complexity of the algorithm is polynomial in the number of CPs, which implies a scalable application of the proposed approach in real scenarios.

The solutions proposed in reference [116] and reference [10] are presented as mechanisms to allow the ISP to perform optimal caching in a scenario of all-encrypted web. We do a step forward by showing that they can represent important building blocks toward the realization of a NN-compliant in-network caching architecture. In the next Chapter, we describe our implementation of a privacy-preserving protocol aimed at enabling a fully-cooperative scheme based on the popularity-based subdivision of the available cache storage. The protocol is designed based on intuitions grasped from both reference [116] and reference [10]. In particular, from [116] we take the idea of associating to each content name a fictitious identifier. Differently from [116], however, our identifiers are not

readable by the ISP but, instead, are ciphered using homomorphic encryption techniques that enable the ISP to apply several meaningful operations on them. From [10] we take the idea of dynamically dividing the available cache storage proportionally to the popularity of the CPs and to make them remotely manage their portion of storage.

3.4 Concluding Remarks

3.4.1 Summary

In this Chapter, we elaborated on Network Neutrality issues that arise from the application of caching strategies. Specifically, we have shown that several characteristics of today's Internet (namely, the wide use of encryption and virtualization) open the doors for cooperative schemes that CPs and ISPs realize to improve the effectiveness of caching. By performing experiments on a simple content delivery scenario, we have shown that CPs' and ISP's caching performance are strongly affected by the employed cooperative scheme. Based on the obtained results, we have concluded that the most suitable approach in a NN scenario is the fully-cooperative. According to this approach, in fact, each CP receives an amount of ISPs' cache storage proportional to the popularity of its contents. The popularity-based cache storage subdivision allows to balance the interest of the ISP (which minimizes its network resource occupation when the most popular contents are cached) and the requirements of NN. The considered idea of NN, in fact, prescribes that the cooperation between ISPs and CPs should be performed according to some objective principle and not based on arbitrary forms of agreements.

3.4.2 Final Comments

The implementation of a NN-compliant caching scheme that we propose in this Chapter is based on the idea that caches' storage should be divided among the CPs proportionally to the popularity of their contents. Whilst being an effective solution for the ISP (as it allows to maximize its Hit-Rate), this approach is privacy-intrusive, since its enforcement requires CPs and ISP to exchange information that are deemed confidential, such as the popularity of CPs' contents and the sizes of ISPs' caches. To address this issue, in the next Chapter we describe our proposal of a NN-compliant protocol to enable ISP and CPs to perform such popularity-driven cache storage subdivision in a privacy-preserving manner.

A Privacy-Preserving Protocol for Network-Neutrality-Compliant Caching in ISP Networks 4

In the previous Chapter, we have shown that the wide use of encryption strategies forces ISPs and CPs to implement cooperative schemes to apply effective caching solutions. This fact raises issues in relation to the neutrality of the ISPs that own the caching systems. In particular, we have shown that different cooperative schemes lead to a different treatment of traffic and, for this reason, caching may lead to discrimination towards the CPs. We then provided a possible definition of NN-compliant caching as the process in which CPs are allocated a portion of cache storage proportional to the popularity of their contents. The application of this popularity-driven cache storage subdivision requires CPs and ISPs to exchange with each other sensitive information, such as contents' popularity and caches' sizes. In this Chapter, we describe a protocol that we propose to make an ISP able to compute this type of subdivision in a privacy-preserving manner, and we show its effectiveness by means of extensive simulations.

4.1 Motivation

A possible view of NN-compliant caching [84, 6] requires the ISP to reserve CPs portions of storage proportional to their contents' popularity. A high-level representation of this concept is depicted in Fig. 4.1. An example of this type of subdivision, that we refer to as *popularity-driven*, is the following: given a cache server that can store an average number of 1000 contents, a CP that owns 500 out the 1000 most requested contents from the users of the ISP is assigned 50% of the available cache storage. We consider the popularity-driven subdivision to be both NN-compliant and effective [7, 6]. In fact, it is NN-compliant since it guarantees the CPs a neutral treatment (because the storage is assigned only based on their attractiveness and not on arbitrary forms of agreement

with the ISP), but also effective for the ISP, that experiences the highest reduction of its network resource occupation when the most popular contents are served directly from its area. In the previous Chapter, we presented a numerical analysis that shows how caching can be discriminatory towards the CPs. We have also shown that the ISP benefits from the application of a popularity-driven subdivision, which seems to imply that ISPs are incentivized to behave in a NN manner. However, compliance to such NN principles would require the ISP to obtain information about contents' popularity, which is unlikely as CPs are increasingly encrypting their contents to protect users' privacy. To cope with this issue, we propose a privacy-preserving protocol by which the ISP can divide its cache storage proportionally to the popularities of CPs' contents. The protocol is based on the Shamir Secret Sharing (SSS) scheme and it is designed to protect both CPs' and ISP's privacy requirements, as ISP and CPs are not required to exchange with each other sensitive information (e.g., the capacities of the caches and the popularities of contents). We assume that a regulator authority (RA) is introduced in the execution of the protocol to ensure that each CP is assigned a fair amount of storage and that no illicit collusion occurs among CPs and ISP. We measure the effectiveness of the proposed protocol by comparing it with two baseline approaches, namely the (i) *static* and the (ii) *network-resource-driven* subdivisions. In the former, each CP receives the same amount of cache storage; in the latter a CP receives a portion of storage proportional to the resource occupation generated by its traffic inside the network of the ISP. Notice that both these approaches adhere to the idea that a non-discriminatory caching is the one that treats the CPs based on some objective criterion instead of arbitrary forms of agreements. To perform the comparison, we develop a discrete-event-based simulator for dynamic VoD traffic provisioning. We measure the performance of the various approaches in terms of the network *Resource Occupation* (RO) measured by the ISP, and the *Hit-Rate* measured by the ISP and CPs. The obtained results show that our proposed protocol allows minimizing network RO and maximizing the overall hit-ratio with respect to baseline approaches and, unlike the baselines, it also guarantees a NN-compliant storage subdivision. Finally, we evaluate the overhead the protocol introduces, which may be considered negligible compared to the reduction of RO that the protocol provides.

4.2 Background

4.2.1 Paillier cryptosystem

Paillier [93] is a type of asymmetric cryptosystem, whose public and private keys are referred in the rest of the Chapter to as pub_k and $priv_k$, respectively. Paillier has additive homomorphic properties, i.e., the summation of two (or more) cypertexts is the encryption of the summation of the relative plaintexts. For example, given two pairs of plaintexts m_1, m_2 and the relative ciphertexts $c_1 = Enc(m_1, pub_k)$, $c_2 = Enc(m_2, pub_k)$, it holds that $m = Dec(c, priv_k)$, where $m = m_1 + m_2$ and $c = c_1 + c_2$.

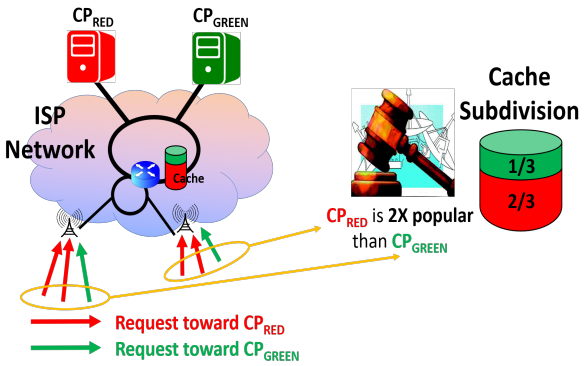


FIGURE 4.1: High-Level representation of the proposed idea of NN-compliant caching: CPs are entitled to receive a portion of storage proportional to their popularity

4.2.2 Shamir Secret Sharing

A (W, T) Shamir Secret Sharing (SSS) scheme [108] allows to share a secret s among a set of W participants in such a way that its reconstruction can only be performed by the collusion of any subset of at least T participants. In the rest of the Chapter, we use the notation $[\![c]\!]_P$ to indicate the share of s assigned to the participant P. We refer the reader to Section 1.3.1 for a description of the atomic operations that can be executed on the shares homomorphically (i.e., to obtain the same results on the corresponding secrets), namely the summation (which is natively implementable with SSS) and the multiplication (for which an additional protocol is required).

4.2.3 Protocol building blocks

During the execution of the protocol, the operations performed over secrets shared with SSS are based on three main atomic operators, namely the *equality-test*, the *comparison* and the *multiplication*. The equality-test (resp., comparison) operator takes as input the shares $[\![s_1]\!]$ and $[\![s_2]\!]$ and returns $[\![1]\!]$ if $s_1 = s_2$ (resp., $s_1 \leq s_2$) and $[\![0]\!]$ otherwise. The multiplication takes as input $[\![s_1]\!]$ and $[\![s_2]\!]$ and returns $[\![s_1 \cdot s_2]\!]$.

As for the equality-test, we employ the equality-test without bit decomposition described in [112]. The equality-test operator serves as the main building block for the implementation of the *aggregate-if-equal* algorithm [66], that takes in input $([\![s_1]\!], [\![v_1]\!])$ and $([\![s_2]\!], [\![v_2]\!])$, i.e., two pairs of (secret,value) in secret shared form and returns $([\![s_1]\!], [\![v_1 + v_2]\!])$ and $([\![s_2]\!], [\![0]\!])$ if $s_1 = s_2$, whereas the pairs are left unchanged otherwise. In our work, we employ another algorithm presented in [66] that efficiently aggregates a sequence of M (secret, value) pairs in secret shared form by recursively applying the *aggregate-if-equal* algorithm for $M \log M$ times.

Our protocol requires to perform several multiplications of secrets. However, the SSS is not homomorphic with respect to the multiplication (i.e., given the shares of two secrets s_1 and s_2,

$[\![s_1 \cdot s_2]\!] \neq [\![s_1]\!] \cdot [\![s_2]\!]$). To address this issue, our protocol exploits the multiplication scheme proposed in [19]. This scheme requires the parties involved in the multiplication to share with each other a multiplicative triple $[\![a]\!], [\![b]\!], [\![c]\!]$ such that $a \cdot b = c$. The security of the multiplication scheme described in [19] is based on the assumption that none of the involved parties is able to obtain the secrets a, b, c from the relative shares $[\![a]\!], [\![b]\!], [\![c]\!]$. The shares of the multiplication triple can be pre-computed in a secure manner using the scheme proposed in [37].

4.3 NN-compliant caching

4.3.1 Definition

In the previous Chapter we advocated the inclusion of caching in the current debate about NN and provided guidelines to reach a possible definition of NN-compliant caching. In this work, we consider caching to be network-neutral if the available ISP's cache storage is divided among the CPs proportionally to the popularity of their contents. This definition allows to balance the requirements of NN (i.e., CPs are treated based on an unbiased criterion instead of on arbitrary forms of agreements) and the legitimate interests of the ISP, which is willing to minimize its network RO in return of the monetary investment done to buy and maintain the caching system [12]. In the next subsection, we formally present the problem of computing a popularity-driven subdivision of the storage, and we briefly introduce the protocol proposed to solve it in a privacy-preserving manner.

4.3.2 Problem Statement

We consider a scenario where a sequence of N requests $\mathcal{R} = \{r_1, r_2, ..., r_N\}$ is issued from the users within the area of an ISP towards a set of K CPs. The ISP owns a caching system composed of several cache servers. The generic n-th cache is characterized by the size of its storage $S^{(n)}_{cache}$, expressed in bytes, and by an average number of contents that it can store, i.e., $N^{(n)}_{cache}$. Assuming that the average size of the contents is $\hat{s}$, it is possible to derive $N^{(n)}_{cache}$ as $N^{(n)}_{cache} = \lfloor \frac{S^{(n)}_{cache}}{\hat{s}} \rfloor$.

Following the definition of NN-compliant caching given in Section 4.3.1, the total storage of the n-th cache (i.e., $S^{(n)}_{cache}$) should be divided among the K CPs proportionally to the popularity of their contents. Specifically, if the n-th cache can store, on average, $N^{(n)}_{cache}$ contents, the k-th CP is entitled to receive a percentage of the total storage proportional to the number of its contents belonging to the $N^{(n)}_{cache}$ most requested contents from the area of the ISP. The portion of storage $\gamma^{(n)}_k$ that the k-th CP is worth receiving is computed as:

$$\gamma^{(n)}_k = \frac{z_k}{\sum_{j=1}^{K} z_j} \cdot S^{(n)}_{cache}, 1 \leq k \leq K \tag{4.1}$$

where $S_{cache}^{(n)}$ is the size of the n-th cache, while z_j is the number of contents offered by the j-th CP whose popularity rank is below $N_{cache}^{(n)}$ (we recall that the most popular content has rank equal to 0).

To compute $\gamma_k^{(n)}, 1 \leq k \leq K$, the ISP and the CPs are required to exchange with each other information that are deemed confidential, such as the size of the caches and the popularity of the contents. The protocol that we propose allows to perform this computation in a privacy-preserving manner. In the next Section, we describe the roles and objectives of the entities involved in the execution of the protocol.

4.4 Architecture

The proposed protocol is executed by three entities, namely an ISP, the CPs and a Regulator Authority (RA). This Section is devoted to the description of the involved parties, their caching objectives, privacy requirements, and security models.

4.4.1 Internet Service Provider

The ISP provides Internet connectivity to its users and it is the owner of the caching system exploited by the CPs. Concerning the execution of our protocol, it has the following objectives/requirements.

Caching Objectives: the main performance objective of the ISP is the minimization of its overall network resource occupation (RO). RO is defined as the amount of resources occupied to deliver all the requests (more specifically, RO is the product of the number of network links traversed by the duration of a request by the bit-rate of the requested content).

Privacy Requirements: ISPs commonly consider confidential the information related to their infrastructure [30]. In this work, we assume that the ISP is not willing to disclose the size of its cache servers, as this may provide precious information on its monetary investment [14]. More specifically, the RA and all the K CPs should not learn any information about the size of the n-th cache (i.e., $S_{cache}^{(n)}$). $CP_k, 1 \leq k \leq K$ can learn, at most, a lower bound $\gamma_k^{(n)}$, which is obtained as a licit output of the protocol. In addition, the RA should not learn $\gamma_k^{(n)}, 1 \leq k \leq K$.

Security Model: we model the ISP as an *honest-but-curious* entity, that executes the protocol truthfully but tries to obtain as many information as possible from its transcripts (e.g., the ISP may try to infer the popularity of a content from the secrets' shares that it receives). A variation of the protocol that can deal with a dishonest ISP (i.e., an ISP that lies in its inputs) is described in Section 4.6, where we present a subprotocol managed by the RA to perform anti-cheating operations.

4.4.2 Content Providers

We consider K CPs, referred to as $CP_k, 1 \leq k \leq K$. A generic CP_k offers a catalogue of contents $\mathcal{C}_k$, which is assumed to be completely stored on a datacenter located outside the area of the ISP.

As proposed in [10], each CP remotely manages its portion of cache storage (e.g., by selecting the contents to be cached) and directly serves its users from the cache. Without loss of generality, we assume that the catalogues of the K CPs do not have any content in common (i.e., single catalogues' entries do not overlap). Moreover, we assume that the catalogues of the K CPs are not equally attractive towards the users, i.e., some catalogues are much more popular than others [48] and that users can retrieve contents from any of such catalogues. We refer to the overall catalogue (i.e., the composition of all the CPs catalogues) to as $\mathcal{C}$.

Caching Objectives: a CP aims to maximize its personal Hit-Rate (i.e., the percentage of requests directed to it that are served from the caches), as this results in an improvement of the overall QoE that it can offer to its users [71].

Privacy Requirements: we assume that the CPs aim to protect the following information:

1. *Confidentiality of the requests*: given the generic request r issued by user u toward CP_k, the ISP, the RA and all the CPs (except CP_k itself) should not be able to identify the requested content with non-neglibigle probability.

2. *Contents' popularity*: given two contents c_x and c_y, the ISP, the RA and all the CPs should not be able to say if c_x is more popular than c_y with non-negligible probability. In case both c_x and c_y belong to the generic CP_k, only that CP can know which content is more popular than the other. It is important to remark that disclosing the information about contents' popularity would reveal extremely confidential insights about the competition between the CPs (e.g., how the market shares are distributed among the CPs).

3. *Number of contents and their size*: the ISP, the RA and all the CPs should not be able to discover the total number of contents owned by the CPs, as well as their sizes.

Security Model: our protocol guarantees a popularity-driven subdivision of the storage, but its effectiveness is based on the assumption that CPs honestly execute it. In fact, if CPs altered their data during the execution of the protocol (e.g., by lying about a requested content), the obtained subdivision would not reflect the correct proportion among CPs' popularity. Driven by the idea that each CP has scarce knowledge about the popularity patterns of the competitors, we assume that it is also not able to alter its data in such a way to obtain a portion of cache storage larger that what it is entitled to receive. Moreover, we assume that the CPs do not have the economical incentives to collude with each other. Hence, CPs can be considered *honest*.

4.4.3 Regulator Authority

The Regulator Authority (RA) is considered a *honest* entity that engages with the ISP and the CPs only the legitimate exchange of information envisioned by the protocol. The RA has the main objective of ensuring a NN-compliant storage subdivision (i.e., popularity-driven) division and acts

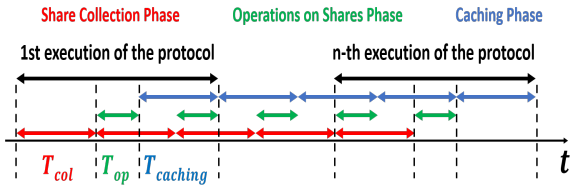


FIGURE 4.2: Phases of the execution of the NN-compliant protocol

as a guarantor that CPs' and ISP's privacy is not violated. Moreover, the RA is the only entity that knows the private key that can be used to decrypt the data ciphered with the Paillier cryptosystem.

4.5 The NN-compliant protocol

The NN-compliant protocol involves a set of operations that are mainly performed over the shares of the secrets that ISP, CPs and RA generate using SSS and exchange among each other. We consider a (2,2) SSS, i.e., only a collusion of 2 out of 2 participants allows to reconstruct the secrets.

The protocol works in four main phases: *preliminary operations*, *share collection*, *operations on shares* and *caching*. The preliminary operations are needed to make the parties learn data (e.g., the shares of the multiplication triples) that will be needed during the execution of the protocol. Hence, such operations can be performed in an off-line fashion. The successive three phases last for a period of T_{col}, T_{op} and $T_{caching}$, respectively, and are cyclically repeated as depicted in Figure 4.2. In the same figure it is also possible to notice that the share collection and the operations on shares phases start simultaneously after the end of the previous share collection phase and that, by construction, $T_{col} = T_{caching}$. We describe the aforementioned phases in the following subsections.

4.5.1 Preliminary operations

Preliminary operation aim is to give the ISP the information on the average size of CPs' contents and to compute the shares of the multiplication triples required to perform secret multiplications.

4.5.1.1 Secure Computation of the Average Dimension of the Contents

First, the ISP learns $\hat{s}$, i.e., the average size of the contents owned by the CPs. This value is needed to obtain the average number of contents that a cache can store (i.e., N_{cache}) from its size S_{cache} (see Eq. 4.1). This phase is designed to allow the CPs to not disclose to the ISP neither the number nor the size of their contents. The k-th CP uses the public key pub_k to encrypt (i) the sum of the sizes of its contents (i.e., $\mathcal{S}_k$) and (ii) the number of contents of its catalogue (i.e., $\mathcal{N}_k$) by means of the Paillier cryptosystem briefly reviewed in Section 4.2.1. Both $Enc(\mathcal{S}_k)$ and $Enc(\mathcal{N}_k)$ are sent to the RA. This operation is performed by all the K CPs. Then, the RA computes $\sum_{k=1}^{K} Enc(\mathcal{N}_k)$ and $\sum_{k=1}^{K} Enc(\mathcal{S}_k)$ that, due to the additive homomorphic properties of the Paillier cryptosystem,

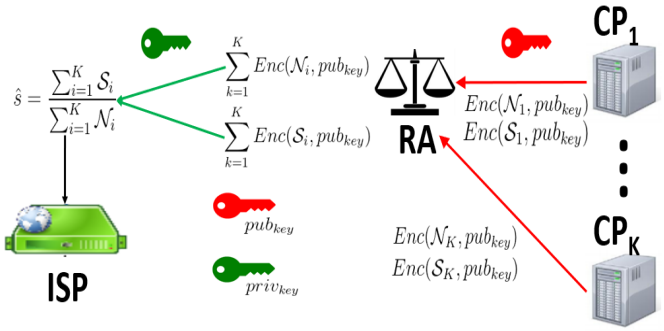


Figure 4.3: Secure computation of the average contents' size

correspond to the encryption of the total number of contents and to the overall summation of their sizes, respectively. The RA, which is assumed to be the only entity who knows the private key $priv_k$, successively decrypts the two values and obtain $\sum_{k=1}^{K} \mathcal{N}_k$ and $\sum_{k=1}^{K} \mathcal{S}_k$. From these values, it is then simple to compute the average size of the contents as $\hat{s} = \frac{\sum_{k=1}^{K} \mathcal{S}_k}{\sum_{k=1}^{K} \mathcal{N}_k}$, which is sent by the RA to the ISP. A representation of this phase is depicted in Fig. 4.3.

4.5.1.2 Secure Computation of a Multiplication Triple

The ISP and the RA compute the shares of a multiplicative triple ($[\![a]\!]_{ISP}$, $[\![a]\!]_{RA}$,$[\![b]\!]_{ISP}$, $[\![b]\!]_{RA}$,$[\![c]\!]_{ISP}$, $[\![c]\!]_{RA}$ such that $c = a \cdot b$) by means of the scheme presented in [37] and briefly reviewed in Section 4.2.3.

4.5.2 Collection of the shares

Upon a new request (say r_i) for content $c_j \in \mathcal{C}$, the owner CP generates two shares of the identifier (e.g., the name) of content c_j, i.e., $[\![r_i]\!]_{ISP} = [\![c_j]\!]_{ISP}$ and $[\![r_i]\!]_{RA} = [\![c_j]\!]_{RA}$, and sends them to the ISP and the RA, respectively. We assume that the ISP and the RA can always associate a share with the owner CP (e.g., by means of its IP address). At the end of this phase, the ISP and the RA know the shares of all the requests $\mathcal{R}$ issued during the share collection phase, i.e., $\mathcal{S}_{ISP} = \{[\![r_i]\!]_{ISP}\}$ and $\mathcal{S}_{RA} = \{[\![r_i]\!]_{RA}\}$, $\forall r_i \in \mathcal{R}$. Notice that, even if the same content c_j is requested in both r_1 and r_2, it holds that $[\![r_1]\!] \neq [\![r_2]\!]$, which prevents the ISP from inferring the popularity patterns of the CPs. The operations performed in this phase are shown in Subprotocol 1.

4.5.3 Operations on shares

Since the ISP and the RA perform the same operations on their set of shares, we omit the apex unless necessary, and we describe the operations performed over the abstract set of shares

Subprotocol 1 Collecting the shares of the requested contents' identifiers

Input: RA: None
ISP: None
CPs: Each CP_k inputs the subset of contents' requests $\mathcal{R} = \{r_1, ..., r_N\}$ directed to it

Output: RA learns $[\![r_i]\!]_{RA}, 1 \leq i \leq N$
ISP learns $[\![r_i]\!]_{ISP}, 1 \leq i \leq N$
CPs learn nothing

1: **for** $1 \leq i \leq N$ **do**
2: Let CP_k be the owner of the content requested in r_i
3: CP_k generates $[\![r_i]\!]_{ISP}$ and $[\![r_i]\!]_{RA}$
4: $CP_k \rightarrow$ RA: $[\![r_i]\!]_{RA}$
5: $CP_k \rightarrow$ ISP: $[\![r_i]\!]_{ISP}$
6: **end for**

$\mathcal{S} = \{[\![r_i]\!], \forall r_i \in \mathcal{R}\}$ that have been collected during the collection phase. The operations performed over the shares are shown in Subprotocol 2 and described in the following:

Subprotocol 2 Performing operations on the shares

Input: RA: $[\![r_i]\!]_{RA}, 1 \leq i \leq N$
ISP: $[\![r_i]\!]_{ISP}, 1 \leq i \leq N$

Output: RA learns $[\![z_k]\!]_{RA}, 1 \leq k \leq K$
ISP learns $[\![z_k]\!]_{ISP}, 1 \leq k \leq K$

1: RA generates two shares of the constant 1: $[\![1]\!]_{RA}$ and $[\![1]\!]_{ISP}$
2: ISP generates two shares of the constant N_{cache}: $[\![N_{cache}]\!]_{RA}$ and $[\![N_{cache}]\!]_{ISP}$
3: RA $\rightarrow$ ISP: $[\![1]\!]_{ISP}$
4: ISP $\rightarrow$ RA: $[\![N_{cache}]\!]_{RA}$

RA and ISP locally execute for j=RA and j=ISP, respectively

1: Execute the *aggregate-if-equal* algorithm over the set $\{([\![r_i]\!]_j, [\![1]\!]_j), 1 \leq i \leq N\}$ and obtain $\{[\![n_i]\!]_j, 1 \leq i \leq N\}$
2: Execute the *comparison* algorithm between all the pairs of elements of the set $\{[\![n_i]\!]_j, 1 \leq i \leq N\}$ and obtain $\{[\![\pi_i]\!]_j, 1 \leq i \leq N\}$
3: **for** $1 \leq i \leq N$ **do**
4: Execute the *comparison* algorithm on $([\![\pi_i]\!]_j, [\![N_{cache}]\!]_j)$ and obtain $[\![\beta_i]\!]_j$
5: The CP_k to which r_i is directed is identified
6: Updating of the number of contents belonging to CP_k whose popularity rank $\pi < N_{cache}$: $[\![z_k]\!]_j \leftarrow [\![z_k]\!]_j + [\![\beta_i]\!]_j$
7: **end for**

4.5.3.1 Aggregate if equal

Given a set $\mathcal{S} = \{[\![r_i]\!], \forall i \in \mathcal{R}\}$ containing the shares relative to the contents of N requests, the objective of this phase is to obtain the share $[\![n_i]\!], \forall i \in \{1, ..., N\}$, where n_i is total number of requests of the content requested in r_i. To perform this operation, we employ the algorithm presented in [66]

and briefly reviewed in Section 4.2, that computes the aggregation of a set of N elements (in the form of secret shares of key and value) by recursively executing the *aggregate-if-equal* algorithm $N \log N$ times. In our application of the protocol, the key is the share of the content c_j hidden in the i-th request r_i, i.e., $[\![r_i]\!] = [\![c_j]\!]$, while the value associated is the share of 1 for all the requests. Since both the ISP and the CPs might be interested in altering the value (as this would favour some contents over others and ultimately affect the caching process), we mandate the RA to generate $[\![1]\!]_{RA}$ and $[\![1]\!]_{ISP}$ at each request. At the end of this phase, the ISP and the RA obtain the respective shares of $[\![n_i]\!], \forall i \in \{1, ..., N\}$.

4.5.3.2 Rank computation

From the previous phase, ISP and RA have obtained a set $[\![n_i]\!], \forall i \in \{1, ..., N\}$ containing the shares of the number of occurrences for each requested content. With these data in hand, they aim at computing $\pi_i \in [0, N-1]$, i.e., the rank of the content requested in r_i, where π_i increases with decreasing popularity of the associated content (i.e., $\pi = 0$ for the most popular content).

To perform this task, all the shares $[\![n_i]\!], \forall i \in \{1, ..., N\}$ need to be compared with each other, for a total of N^2 executions of the *comparison* algorithm mentioned in Section 4.2. We recall that the algorithm takes in input two shares $[\![x_1]\!]$ and $[\![x_2]\!]$ and returns $[\![1]\!]$ if $x_1 \leq x_2$, and $[\![0]\!]$ otherwise. Considering that $[\![1]\!] = 1 - [\![0]\!]$ and $[\![0]\!] = 1 - [\![1]\!]$ (due to the additive homomorphic properties of SSS), it is possible to assign the share $[\![1]\!]$ to the lower value (say x_1) and the share $[\![0]\!]$ to the higher one (say x_2) with a single execution of the comparison algorithm. Hence, the complexity is reduced from N^2 to $\binom{N}{2}$ executions of the *comparison* algorithm. The rank π_i can then be computed by summing up the results, in secret shared form, of the relative comparisons as $[\![\pi_i]\!] = \sum_{x=1, x \neq i}^{N-1} [\![l_{i,x}]\!]$, where $l_{i,x} = 0$ if $n_i \leq n_x$ (and 1 otherwise). Notice that, if r_i is the request relative to the most popular content, then $l_{i,x} = 0, \forall x$ (because its number of occurrences is higher than all the others) and, as expected, $\pi_i = 0$.

Once the shares of ranks $[\![\pi_i]\!], \forall i \in \{1, ..., N\}$ have been obtained, ISP and RA need to compute the share of the portion of cache that each CP is expected to receive. To this aim, the rank of each content needs to be compared with the size of the cache and, if $\pi_i \leq N_{cache}$, the CP to which r_i is directed is entitled to store one content. Since the ISP wants to protect the information about its cache size, it generates 2 shares $[\![N_{cache}]\!]_{ISP}$ and $[\![N_{cache}]\!]_{RA}$, which can be used to perform a comparison with $[\![\pi_i]\!], \forall i \in [1, ..., N]$ by means of the *comparison* algorithm. The result of the i-th comparison is $[\![\beta_i]\!]$, with $\beta_i = 1$ if $\pi_i \leq N_{cache}$ (and 0 otherwise).

By repeating this operation for all the requests directed towards CP_k, i.e., $\mathcal{R}_k$, ISP and RA obtain the share of the number of most popular contents owned by CP_k as $[\![z_k]\!] = \sum_{i \in \mathcal{R}_k} [\![\beta_i]\!]$.

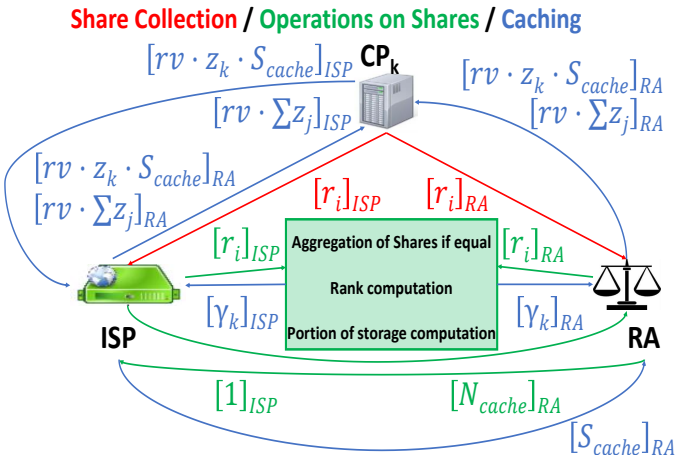


FIGURE 4.4: Main shares learnt by ISP, CPs and RA during the execution of the NN-compliant protocol

4.5.4 Caching

However, we prevent the ISP and the RA from directly reconstructing these secrets since (i) from $\sum_{j=1}^{K} z_j$ the RA could obtain a good estimate of the size of the cache S_{cache} and (ii) from $z_k, 1 \leq k \leq K$ it would discover the number of contents owned by each CP whose popularity rank is less than N_c. Instead, ISP and RA employ the scheme described in [112] to obtain the shares of a random integer $[\![rv]\!]_{ISP}$ and $[\![rv]\!]_{RA}$ without learning rv itself. With these values in hands, they then learn, using the multiplicative protocol proposed in [19], $[\![rv \cdot z_k \cdot S_{cache}]\!], 1 \leq k \leq K$ and $[\![rv \cdot \sum_{j=1}^{K} z_j]\!]$. Notice that these values represent the shares of the numerator and denominator of $\frac{z_k}{\sum_{j=1}^{K} z_j}$, respectively, which have been masked with the same value rv to keep the ratio between them unchanged (and equal to γ_k).

Then, the RA sends $[\![rv \cdot z_k \cdot S_{cache}]\!]_{RA}$ and $[\![rv \cdot \sum_{j=1}^{K} z_j]\!]_{RA}$ to the corresponding k-th CP, which exchanges with the ISP their shares to recover $rv \cdot z_k \cdot S_{cache}$ and $rv \cdot \sum_{j=1}^{K} z_j$. From these two reconstructed secrets, both the ISP and the k-th CP compute the amount of storage destined to CP_k:

$$\gamma_k = \frac{rv \cdot z_k \cdot S_{cache}}{rv \cdot \sum_{j=1}^{K} z_j} \tag{4.2}$$

Notice that the k-th CP learns nothing more that its allocated storage. For example, it does not learn the percentage of storage it is assigned, from which it would have derived the size of the cache S_{cache}.

At this point, the k-th CP can start caching its contents in the received storage portion. Notice also that, whilst the popularity-based caches' subdivision is computed by performing operations on the shares relative to contents' requests (i.e., the proposed protocol is designed to work at the

Subprotocol 3 Calculating the portion of cache storage to allocate to each CP

Input: RA: $[\![S_{cache}]\!]_{RA}, [\![z_k]\!]_{RA}, 1 \leq k \leq K$
ISP: $[\![S_{cache}]\!]_{ISP}, [\![z_k]\!]_{ISP}, 1 \leq k \leq K$
CPs: None

Output: RA learns nothing
ISP learns $\gamma_k = \frac{rv \cdot z_k \cdot S_{cache}}{rv \cdot \sum_{j=1}^{K} z_j}, 1 \leq k \leq K$
Each CP_k learns $\gamma_k = \frac{rv \cdot z_k \cdot S_{cache}}{rv \cdot \sum_{j=1}^{K} z_j}$

1: ISP and RA compute the shares $[\![rv]\!]_{ISP}$ and $[\![rv]\!]_{RA}$ of a secret random integer rv using the scheme of [112]
2: ISP and RA compute $[\![rv \cdot \sum_{j=1}^{K} z_j]\!]_{ISP}$ and $[\![rv \cdot \sum_{j=1}^{K} z_j]\!]_{RA}$ using the multiplication protocol of [19]
3: **for** $k \in \{1, ..., K\}$ **do**
4: RA $\rightarrow CP_k$: $[\![rv \cdot \sum_{j=1}^{K} z_j]\!]_{RA}$
5: ISP computes $[\![rv \cdot z_k \cdot S_{cache}]\!]_{ISP}$
6: RA computes $[\![rv \cdot z_k \cdot S_{cache}]\!]_{RA}$
7: RA $\rightarrow CP_k$: $[\![rv \cdot z_k \cdot S_{cache}]\!]_{RA}$
8: **end for**
9: **for** $k \in \{1, ..., K\}$ **do**
10: $CP_k \rightarrow$ ISP: $[\![rv \cdot \sum_{j=1}^{K} z_j]\!]_{RA}$ and $[\![rv \cdot z_k \cdot S_{cache}]\!]_{RA}$
11: ISP $\rightarrow CP_k$: $[\![rv \cdot \sum_{j=1}^{K} z_j]\!]_{ISP}$ and $[\![rv \cdot z_k \cdot S_{cache}]\!]_{ISP}$
12: **end for**
13: **for** $k \in \{1, ..., K\}$ **do**
14: ISP and CP_k reconstruct the secret $rv \cdot \sum_{j=1}^{K} z_j \leftarrow \left([\![rv \cdot \sum_{j=1}^{K} z_j]\!]_{RA}, [\![rv \cdot \sum_{j=1}^{K} z_j]\!]_{ISP}\right)$
15: ISP and CP_k reconstruct the secret $rv \cdot z_k \cdot S_{cache} \leftarrow ([\![rv \cdot z_k \cdot S_{cache}]\!]_{RA}, [\![rv \cdot z_k \cdot S_{cache}]\!]_{ISP})$
16: **end for**
17: **for** $k \in \{1, ..., K\}$ **do**
18: ISP computes $\gamma_k = \frac{rv \cdot z_k \cdot S_{cache}}{rv \cdot \sum_{j=1}^{K} z_j}$
19: CP_k computes $\gamma_k = \frac{rv \cdot z_k \cdot S_{cache}}{rv \cdot \sum_{j=1}^{K} z_j}$
20: **end for**

content level) caching strategies are successively applied by the CPs on a chunk-level basis, as further described in Section 4.7.2. In Fig. 4.4 we depict the most salient shares that the involved parties exchange with each other during the last three phases of execution of the protocol, namely collection of shares, operations on shares and caching.

We remind that CP_k is entitled to receive a portion of storage $\gamma_k = \frac{z_k}{\sum_{j=1}^{K} z_j} \cdot S_{cache}$. The ISP and the RA know their shares $[\![S_{cache}]\!]$ and $[\![z_k]\!], 1 \leq k \leq K$ and could recover z_k, $\sum_{j=1}^{K} z_j$ and S_{cache} and obtain from them the value γ_k. The exchange of shares and the operations performed on them to compute the cache storage subdivision are described in the following and shown in Subprotocol 3.

4.5.5 Fullfilment of Privacy Requirements

4.5.5.1 ISP's Privacy Requirements

We remind that neither the RA nor the CPs are allowed to obtain the size of the ISP's caches (i.e., S_{cache}) and that the RA is not allowed to obtain the portion of storage given to CP_k, i.e., $\gamma_k, 1 \leq k \leq K$.

During the execution of the protocol (in the rank computation phase, precisely) the RA obtains the share $[\![N_{cache}]\!]$. Since SSS is proved secure under the information-theoretic security model [108], this share provides absolutely no additional information on the relative secret. Hence, the RA does not discover the size of the cache. Then, in the caching phase, the RA learns $[\![rv \cdot z_k \cdot S_{cache}]\!]$ and $[\![rv \cdot \sum_{j=1}^{K} z_j]\!]$. Under the assumption of honest RA, ISP and RA do not exchange their shares with each other. Hence, the RA does not obtain γ_k.

During the caching phase, $CP_k, , 1 \leq k \leq K$ learns $rv \cdot z_k \cdot S_{cache}$ and $rv \cdot \sum_{j=1}^{K} z_j$, from which it computes $\gamma_k = \frac{rv \cdot z_k \cdot S_{cache}}{rv \cdot \sum_{j=1}^{K} z_j}$. Notice that the ability of CP_k to estimate S_{cache} is bounded by its ability to assess its popularity with respect to the popularity of its competitors, which is encoded in the ratio $\frac{z_k}{\sum_{j=1}^{K} z_j}$. Since we have assumed that each CP has scarse knowledge about other CPs' attractiveness, we consider S_{cache} to be protected from the CPs as well.

4.5.5.2 CP's Privacy Requirements

Considering the first privacy requirement of the CP, i.e., the confidentiality of the requests, in the share collection phase the ISP and the RA receive $[\![r_i]\!]_{ISP}$ and $[\![r_i]\!]_{RA}$ from the CP towards which the i-th request is issued (say CP_k). As SSS is information-theoretically secure, it holds that:

$$P(c_j|[\![r_i]\!]) = \frac{1}{\mathcal{N}_k}, \quad \forall j \in \{1, ..., \mathcal{N}_k\} \tag{4.3}$$

where $P(c_j|[\![r_i]\!])$ is the probability that the share $[\![r_i]\!]$ refers to content c_j and $\mathcal{N}_k$ is the total number of contents owned by CP_k. No CP (except CP_k) obtains $[\![r_i]\!]$ from the execution of the protocol and both the RA and the ISP can identify the content hidden behind the i-th request with a negligible probability only. Hence, the first CPs' privacy requirement is fulfilled.

Concerning the second privacy requirement, i.e., protection of contents' popularity, in the rank computation phase the ISP and the RA obtain the shares of the number of occurrences of the content hidden behind the i-th request, i.e., $[\![n_i]\!], \quad \forall i$. They then compare the number of occurrences of each pair of contents (say c_i and c_x) and obtain the result in secret shared form (i.e., $[\![l_{i,x}]\!]$). Due to the information-theoretically security properties of SSS, $P(l_{i,x} = 0) = P(l_{i,x} = 1) = 0.5$. Hence, neither the ISP nor the RA can violate the privacy of contents' popularity.

Finally, to satisfy the third privacy requirement, the ISP should not obtain the number and the sizes' of CPs' contents. During the preliminary operations, the ISP only obtains the average size of contents $\hat{s}$, from which it cannot derive neither the total number of contents, nor their sizes.

Table 4.1: Table of Notations

Protocol's variables and relative description	
Variable	**Description**
r_i	i-th request issued from users
$\mathcal{R} = \{r_i\}$	Set of all the requests
$[\![x]\!]$	Share of the generic value x
n_i	Number of requests for the content hidden behind the i-th request
π_i	Popularity rank of the content hidden behind the i-th request
z_k	Number of contents belonging to CP_k whose popularity rank is below N_{cache}
γ_k	Amount of cache storage allocated to CP_k (measured in bytes)
α	Skewness parameter of the contents' popularity distribution
ϕ	Bit-length representation of a share

4.6 Extension of the Protocol for dishonest ISP

In this Section, we describe a scenario in which, by maliciously forging its data, the ISP can obtain an unfair subdivision of the cache storage. We then provide an extension of the protocol to make the RA able to discover if the ISP is cheating.

The generic cache server is characterized by its size S_{cache} and by the average number of contents that it can store N_{cache}, according to the relation $S_{cache} = N_{cache} \cdot \hat{s}$, being $\hat{s}$ the average size of CPs' contents. N_c determines the number of contents that the CPs regard as the most popular ones. Just as an example, let us think of the case of 2 CPs, referred to as CP_1 and CP_2, which own contents whose popularity ranks go from 0 to 49 and from 50 to 99, respectively. If $N_{cache} = 50$ contents, then, according to our definition of NN-compliant caching, the 100% of the total cache storage should be assigned to CP_1. This value drops to 50% if, instead, $N_{cache} = 100$ contents. This scenario shows that, by communicating to the RA the share of a forged N_{cache}, the ISP is able favour a specific CP. To address this issue, the RA can compare $\hat{s} \cdot [\![N_c]\!]_{RA}$ and $[\![S_c]\!]_{RA}$ using the equality-test operator, and ask the ISP to perform a similar operation. By doing so, RA and ISP learn the shares $[\![b_{eq}]\!]_{RA}$ and $[\![b_{eq}]\!]_{ISP}$, from which they recover b_{eq} that is equal to 1 if the ISP did not forged N_{cache}, and 0 otherwise.

Table 4.2: Table of Notations

Protocol's variables and relative description	
Variable	**Description**
r_i	i-th request issued from users
$\mathcal{R} = \{r_i\}$	Set of all the requests
$[\![x]\!]$	Share of the generic value x
n_i	Number of requests for the content hidden behind the i-th request
π_i	Popularity rank of the content hidden behind the i-th request
z_k	Number of contents belonging to CP_k whose popularity rank is below N_{cache}
γ_k	Amount of cache storage allocated to CP_k (measured in bytes)
α	Skewness parameter of the contents' popularity distribution
ϕ	Bit-length representation of a share
Simulation settings's Variables and relative description	
Variable	**Description**
N	Total number of contents' requests
M	Total number of available contents
$\mathcal{N}_k$	Number of contents of CP_k's catalogue
$\mathcal{S}_k$	Sum of the sizes (measured in bytes) of the CP_k's contents
$\hat{s}$	Average size (measured in bytes) of the M available contents
$N_{cache}^{(n)}$	Capacity of the generic n-th ISP's cache (measured as the average number of contents that can be stored on it)
$S_{cache}^{(n)}$	Capacity of the generic n-th ISP's cache (measured in bytes)
α	Skewness parameter of the contents' popularity distribution
ϕ	Bit-length representation of a share

4.7 Dynamic Simulations for VoD Content Caching and Distribution

We perform extensive simulations to evaluate the performance of the proposed privacy-preserving network-neutrality compliant caching protocol. These simulations can be divided into two main groups: in the first one, that we refer to as *simplified simulative scenario*, a simplified version of the protocol is employed and simulations are performed considering a caching system composed of one single cache server. The aim of these simulations is to show the effectiveness of the popularity-based subdivision with respect to the static one (in terms of caching Hit-Ratio) and to assess the scalability of the protocol with increasing number of CPs and volume of their catalogues. This scenario is simplified since (i) all the contents are assumed to be of the same size and the capacity of the cache server N_{cache} is measured in number of contents that it can store (and not in Bytes) and (ii) a simplified version of the protocol is considered. Specifically, by using it, the generic CP_k receives an amount of cache storage equal to the number of its contents whose popularity is less then

the capacity of the cache server, i.e., $\gamma_k = z_k$ (whereas in the full implementation of the protocol CP_k would have received a portion of the total cache storage proportional to the popularity of its contents, i.e., $\gamma_k = \frac{z_k}{\sum_{j=1}^{K} z_j} \cdot S_{cache}$). This simplified approach may lead to an inefficient utilization of the storage, i.e., when the number of requested contents is less then the overall capacity of the cache. In the second group of simulations, that we refer to as *extended simulative scenario*, we develop a discrete-event-driven simulator to perform dynamic simulations of VoD content caching and distribution considering a more complex network of cache servers and three approaches to divide the available storage, namely the popularity-based, the resource-occupation-based and the static one. The aim of these simulations is to show the effectiveness of the popularity-based subdivision against a more challenging baseline (i.e., the resource-occupation-based caches' subdivision) and considering other metrics with respect to the Hit-Ratio, e.g., the occupation of network resources. Indeed, our overall objective is to evaluate the gain given by our protocol (in terms of caching performance) over the baselines, provided that all the considered approaches (i.e., our method and the baselines) are privacy-preserving to the same extent. Notice, in fact, that we selected the baselines in such a way that they do not require any exchange of sensitive data between ISP and CPs.

For both the simplified and extended simulative scenarios, we adopt the same traffic model, which is described in the following Subsection. Then, in Subsection 4.7.2, we describe in detail the simulation process, e.g., we present the developed simulator, the VoD request provisioning process and the general simulation settings. Finally, in Subsection 4.7.3, we provide details on the considered caching systems.

4.7.1 Traffic Model

Information about contents' requests is widely considered sensitive and business relevant by CPs. Hence, public data sets are rarely available to the research community and we had to perform our simulations over synthetic traffic traces, which have been crafted as follows. Based on a common assumption made in the literature, we consider a fixed catalogue [46] of contents whose popularities are distributed according to the Zipf law, i.e., $p_j = \frac{j^{-\alpha}}{\sum_{z=0}^{M-1} p_z}, \forall j \in \{0, ..., M-1\}$, where p_j is the probability that the j-th popular content is chosen among the available M videos. $\alpha \in [0, 1]$ is the skew parameter of the Zipf (the number of scarcely-requested contents increases with increasing α). Inspired by [85], we also introduce a temporal dynamic to this popularity distribution. In particular, every 30 minutes we sum (or subtract, with the same probability) a Poisson-distributed random variable (with mean value 1) to the popularity rank of each content $c_j, \forall j \in \{0, ..., M-1\}$. Notice that the described catalogue results from the aggregation of the single catalogues owned by each CP.

Finally, we consider CPs that offer, on average, contents of significantly different popularities. Although being a well-known characteristic of existing CPs (few of which are much more popular than the others), to our knowledge a contents' popularity model that take this fact into consideration has never been proposed in the literature. To fill this gap, we propose a model that is described in

the following. We assume that the k-th CP is characterized by a gaussian probability distribution $\rho_j^{(k)}$ over the ranks of the overall catalogue with mean value $\mu_k = \frac{M}{K} \cdot (k - \frac{1}{2})$ and standard deviation $\sigma_k = \sigma_1 + (K - k) \cdot \frac{\sigma_2 - \sigma_1}{K}$, where M and K are the total number of contents and of CPs, respectively. σ_1 and σ_2 are tuned to obtain different degrees of CPs' popularity, in particular to model the difference of CPs' attractiveness towards the users.

According to the proposed model, the j-th popular content of the overall catalogue described above belongs to the k-th CP with probability $P(j,k) = \frac{\rho_j^{(k)}}{\sum_{x=1}^{K} \rho_j^{(x)}}$. In this way, for example, given $K = 5$ CPs and $M = 25000$ contents and considering $\sigma_1 = \frac{M}{K}$ and $\sigma_2 = \frac{\sigma_1}{K}$, CP_1 and CP_5 are assigned contents with an average rank of 4369 and 22144, and standard deviations of 3348 and 1946, respectively. This makes the contents offered by CP_5 much less popular than those owned by CP_1, on average.

In the simplified simulative scenario we measure the capacity of the cache server considering the number of contents that it can store and all the contents are assumed to be of the same size. Instead, in the extended simulative scenario, caches' storage capacities are measure in Bytes, and we assume that the duration of the contents is a random variable distributed according to a Pareto distribution with skew parameter equal to 0.25. All the durations are then normalized between 1200 s and 8400 s. We then assume the same bit-representation for all the contents to be equal to 12 Mbits (hence contents have a size that ranges between 1.8 and 12.6 Gbytes).

4.7.2 Dynamic VoD Content Caching and Distribution Simulator

4.7.2.1 Simplified Scenario

In the simplified scenario, we perform caching using the LRU strategy (i.e., the last requested content is inserted in cache). Notice that in this scenario caching is performed on a content-basis, while in the extended one caching is performed on the chunks in which the contents are divided.

4.7.2.2 Extended Scenario

The overall framework of the simulator developed to perform simulations in the extended scenario is described as follows: given the network topology, content catalogue characteristics of each CP, locations of caches and the list of stored contents per CP in each cache, the simulator provisions the dynamically-arriving VoD-content requests, based on current network status, and gives as an output the overall amount of resources occupied to provision contents of a specific CP, the overall RO of the network and caches' hit-ratios.

Note that a VoD-content request is provisioned taking into consideration its chunk-nature, i.e., each VoD request, according to its duration, consists of a number of chunks and the chunks are provisioned sequentially. This allows to have different chunks of the same VoD request delivered from different caches, which is basically the case when caches are dynamically updated, i.e., when

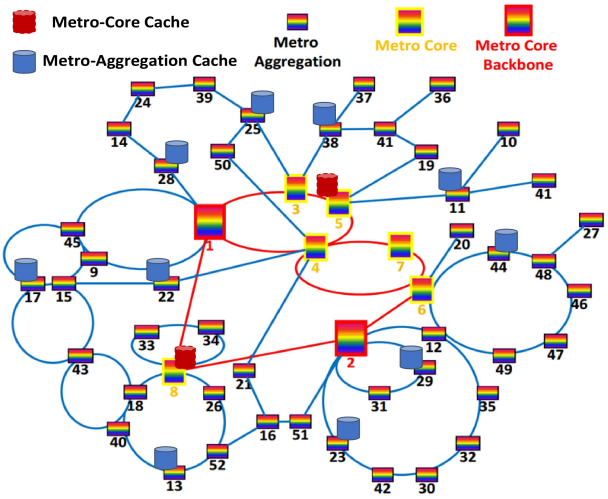


Figure 4.5: Schematic representation of the ISP Network Topology and the location of caches.

contents are pulled out from or pushed in caches. Specifically, a VoD-chunk request is described by the tuple r = (t_r, D_r, b_r, m, d_r), where t_r is the request arriving time from node D_r, b_r is the requested bit-rate, m is the requested content and d_r is the chunk duration. The simulated VoD-chunk provisioning/deprovisioning process is described as follows: Upon arrival of a VoD-chunk request for content m from node D_r, a list of all cache nodes hosting m (including the video server) is identified. Then, the nearest cache storing content m delivers the chunk to node D_r, considering a path with available bandwidth greater than or equal to b_r. The chunk is later deprovisioned at time $t_s + d_r$ deallocating the assigned bandwidth from the utilized path.

4.7.3 Network Model and Caching System

4.7.3.1 Simplified Scenario

We consider one single cache server, whose capacity is measured as the number of contents that it can store.

4.7.3.2 Extended Scenario

We consider a real ISP metro-aggregation network topology, depicted in Fig. 4.5. The network consists of three types of nodes, namely metro-core backbone nodes, metro-core nodes and metro-aggregation nodes. We assume that the metro-core and metro-aggregation nodes are cache-enabled nodes, i.e., capable of hosting and delivering video contents while the metro-core backbone nodes are routers connecting the ISP to the Internet. As for the cache-enabled nodes, we considered 2 metro-core and 12 metro-aggregation caches whose locations are highlighted in Fig. 4.5.

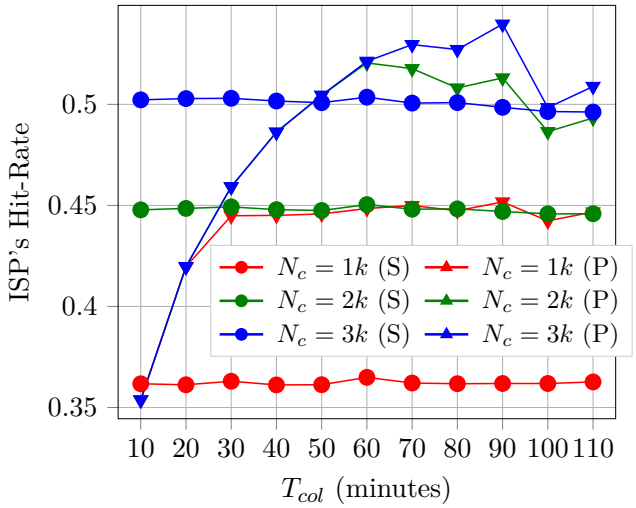


FIGURE 4.6: Comparison of the Hit-Rates experienced with the popularity-driven and the static subdivision with varying T_{col}, $K = 5$ CPs and $\hat{M} = 5000$ average contents per CP. N_c indicates the dimension of the cache (in number of stored contents); S and P stand for *static* and *popularity-driven* cache subdivision strategy, respectively.

4.8 Illustrative Simulative Results

4.8.1 Simplified Simulative Scenario

Our main performance metric is the Hit-Rate experienced by the ISP that divides its cache using our protocol with respect to a static division. We show the average of the results obtained in 100 simulations in which the LRU caching strategy is used. We consider sequences of requests generated according to the traffic model described in Section 4.7.1 and issued with a rate of $\lambda = 1 req/sec$ during a total period of 6 hours (skewness parameter of the Zipf function $\alpha = 0.9$). In Figure 4.8 we depict the Hit-Rate with increasing $T_{col} \in \{10, 20, 30, ..., 110\}$ minutes and $N_{cache} \in \{1000, 2000, 3000\}$ considering $K = 5$ CPs and $\hat{M} = 5000$ contents as average dimension of their catalogues.

Increasing T_{col} has two conflicting objectives on the Hit-Rate relative to the popularity-driven subdivision. From one side, the subdivision of the cache is computed based on more information and can therefore better reflect the actual proportions of popularities; moreover, for low values of T_{col} the cache is not efficiently used since the number of unique contents that are requested is likely to be less than the dimension of the cache (i.e., $\sum_{k=1}^{K} \gamma_k < N_{cache}$). From the other side, the delay between the computation of γ and its actual enforcement increases according to Eq. 4.5 and this may make the obtained γ out-of-date. This results in Hit-Rates that reach their maximum at T_{col} between 60 and 90 minutes and then tend to decrease with increasing T_{col}.

We notice a general gain over the static subdivision, whose performance are, as expected, not dependent from T_{col}. We notice, for example, that our method allows to reach an Hit-Rate $\simeq 0.45$

when $N_{cache} = 1000$, whereas a static division of the cache would require $N_{cache} = 2000$ to guarantee the same performance. The gain decreases with incresing N_{cache}, i.e., when the available storage is enough to accomodate all most popular contents of the CPs.

We then run a second set of experiments with the objective to assess the effect of $N_{cache} \in \{1000, 2000, 3000\}$ on the overall performance changing number of CPs $K \in \{2, 5, 10\}$ and average size of their catalogues $\hat{M} \in \{5000, 10000, 20000\}$. The results shown in Table 4.3 are obtained setting $T_{col} = 80$ minutes (and $T_{op} \simeq 90$ minutes, according to Eq. 4.5). These simulations confirm that the gain becomes more evident when the cache resources are scarce with respect to the number available contents. Moreover, the protocol is demonstrated to scale with respect to both the number of CPs and the size of their catalogues. For example, the gain increases from $\simeq 9\%$ to $\simeq 43\%$ with $N_{cache} = 1000$ when K goes from 2 to 10 CPs.

4.8.2 Extended Simulative Scenario

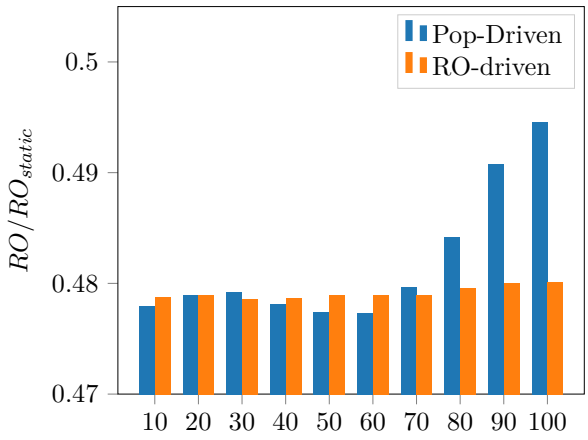
4.8.2.1 Simulation Settings

In this set of experiments, we consider three approaches of cache storage subdivision, namely the popularity-driven, the resource-occupation-driven and the static subdivisions. In the first approach, which is enabled by the use of our protocol, each CP receives a portion of storage proportional to the popularity of its contents. In the second approach each CP receives a portion of storage proportional to the RO the the delivery of its contents generates within the network of the ISP. In the third approach, all the CPs receive the same amount of storage.

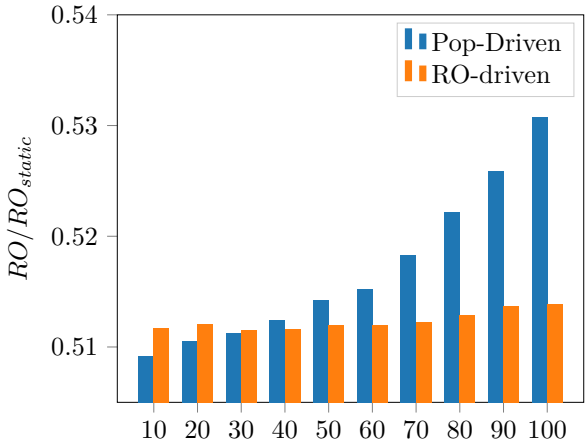
To compare the performance of these approaches, we perform simulations on two different scenarios, the first characterized by $K = 5$ CPs and $\hat{M} = 5000$, and the second by $K = 10$ CPs and $\hat{M} = 5000$, for values of $T_{col} \in \{10, 20, 30, ..., 100\}$ minutes. In each simulation, we simulate the arrival of 43000 VoD requests generated according to the traffic model described in Sec. 4.7.1 at an arrival rate guaranteeing negligible blocking probability (i.e., Zipf $\alpha = 0.8$ and $\lambda = 1 req/sec$), to provide a fair comparative analysis between the considered approaches. We assume the network

Table 4.3: Impact of N_{cache} on the gain with respect to the Hit-Rate with changing number of CPs K and average dimension of their catalogues $\hat{M}$

$\hat{M} = 10^3$		$K = 2$	$K = 5$	$K = 10$
	$N_{cache} = 1k$	9.3%	21.3%	43.1%
	$N_{cache} = 2k$	3.2%	13.9%	28.6%
	$N_{cache} = 3k$	0%	5.45%	20.3%
$K = 5$		$\hat{M} = 5k$	$\hat{M} = 10k$	$\hat{M} = 20k$
	$N_{cache} = 1k$	23.5%	25.2%	26.3%
	$N_{cache} = 2k$	13.9%	16.1%	18.1%
	$N_{cache} = 3k$	5.4%	9.78%	13.3%



(a) *Comparison of the RO considering a scenario with* 5 *CPs with an average number of* 5000 *contents*



(b) *Comparison of the RO considering a scenario with* 10 *CPs with an average number of* 5000 *contents*

FIGURE 4.7: Resource Occupation vs T_{col} obtained with the popularity-driven and with the resource-occupation-driven subdivisions divided by the RO achieved with the static subdivision.

topology shown in Fig. 4.5 with cache locations highlighted. We fix the size of caches located at metro-aggregation nodes and those located at metro-core nodes to 5% and 10% of the overall content catalogues size (of all content catalogues of all CPs).

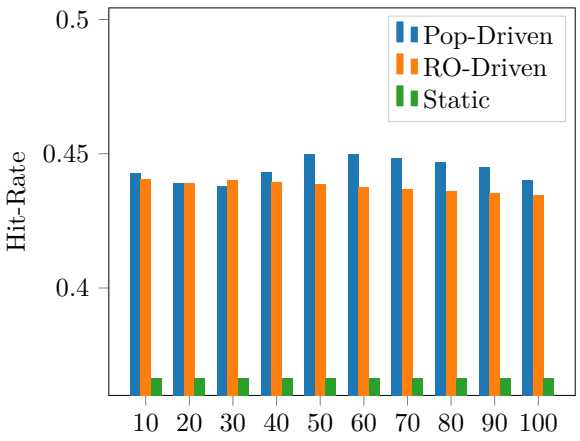
4.8.2.2 ISP's Resource Occupation and Caching Hit-Rate

In this section, we show the comparison of popularity-driven, resource-occupation-driven and static subdivisions considering the overall network RO and the Hit-Rate measured by the ISP for increasing T_{col}. First, we depict the RO obtained with the former approaches as a percentage of the RO measured when the static subdivision is enforced (which is equal to $\sim 760 \cdot 10^6 Mbit$ if $K = 5$ CPs and $\sim 713 \cdot 10^6 Mbit$ if $K = 10$ CPs). The $\frac{RO}{RO_{static}}$ as a function of T_{col} is depicted in Fig. 4.7a and Fig. 4.7b, for the scenarios with 5 and 10 CPs, respectively. We remind that an approach is preferable to the ISP if it reduces the RO measured within its network. We note that both the popularity-driven and the resource-occupation-driven subdivision lead to a remarkable RO gain with respect to the static subdivision. In both the scenarios under analysis, the minimum RO is obtained when the storage of the caches is divided according to the popularity-driven subdivision. More specifically, the minimum RO is obtained with $T_{col} = 10$ minutes and at $T_{col} = 50$ minutes when 5 CPs and 10 CPs are considered, respectively. This result confirms that the effectiveness of caching highly depends on information about contents' popularity and motivates the adoption of our protocol as a tool to keep this information private.

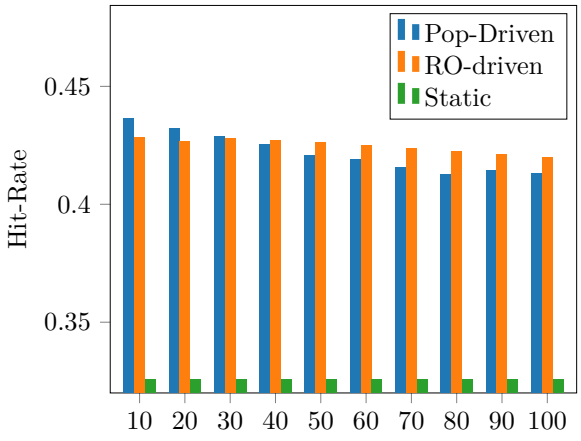
In general, we observe a RO increase for increasing T_{col}. This fact can be explained considering that high values of T_{col} allow the ISP to obtain more information (e.g., about contents' popularity),

but, at the same time, increases the number of changes that contents' popularity undergo during T_{col}. This increase is much more evident in the popularity-driven subdivision, with such percentage going from $\sim 50.9\%$ to $\sim 53\%$ when T_{col} passes from 10 to 100 minutes, whereas the percentage increases only slightly and it is mostly stable around $\sim 51\%$ when the resource-occupation-driven subdivision is employed. This difference between the two approaches is due to fact that our protocol introduces a delay between the computation of the popularity-driven subdivision and its actual enforcement. This delay increases with increasing T_{col} and this may make the computed storage subdivision out-of-date with respect to the current popularity patterns (we elaborate further on the dependency between this delay and T_{col} in Section 4.8.2.4). The conflicting effects of increasing T_{col} are more visible in Fig. 4.7a, where it is possible to observe that the RO of the popularity-driven subdivision decreases until the minimum value is reached (at $T_{col} = 50$ minutes) and then increases up to the maximum (at $T_{col} = 100$ minutes).

Fig. 4.8 shows the Hit-Rates measured at the caches located in the metro-aggregation level (i.e., the percentage of requests served from the caches closer to the users). Obtained results are consistent with the RO previously described: (i) the Hit-Rates of popularity-driven and resource-occupation-driven subdivisions significantly outperform the static subdivision and (ii) the RO decreases (resp., increases) when the Hit-Rate increases (resp., decreases). The maximum Hit-Rates obtained with a popularity-driven subdivision are higher than the benchmarks in both scenarios. For example, in the scenario with 5 CPs, the maximum Hit-Rates for the popularity-driven and for the resource-driven subdivision are ~ 0.45 and ~ 0.44, respectively (see Fig. 4.8a). When instead 10 CPs are considered, the corresponding values are ~ 0.436 and ~ 0.428 (see Fig. 4.8b).

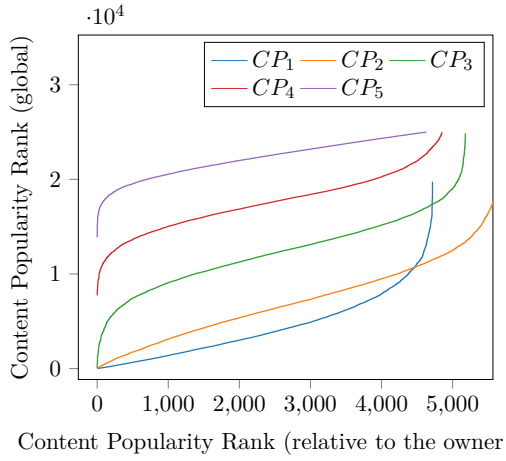


(a) *Comparison of the ISP's Hit-Rate considering a scenario with* 5 *CPs with an average number of contents of* 5000

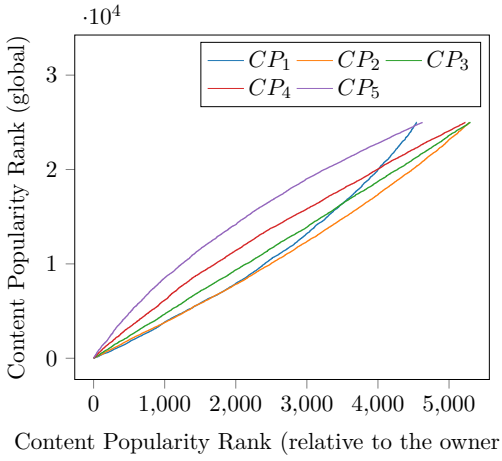


(b) *Comparison of the Hit-Rate considering a scenario with* 10 *CPs with an average number of contents of* 5000

Figure 4.8: ISP's Hit-Rate (measured at metro-aggregation caches) vs T_{col} obtained with the popularity-driven, the resource-occupation-driven and the static subdivisions.



(a) 5 *CPs offering an average number of* 5000 *contents; the attractiveness towards the users is significantly-different among the CPs*



(b) 5 *CPs offering an average number of* 5000 *contents; the attractiveness towards the users is similar among the CPs*

4.8.2.3 Hit-Rates for the CPs

According to our vision of NN an ISP should maximally benefit from the application of caching strategies, as long as they are not discriminatory towards the CPs. Therefore, we believe that the ISP is entitled to decide how frequenty the protocol is executed (i.e., by setting T_{col} to the value that minimizes the RO). However, since the value that minimizes the RO is not necessarily the one that maximizes the hit-ratio of every CP, CPs may experience a loss in their hit-ratio. We formally define this loss as:

$$\mathcal{L}_k = \frac{\hat{h}^{(k)} - \hat{h}^{(k)}_{isp}}{\hat{h}^{(k)}}, 1 \leq k \leq K \tag{4.4}$$

where $\hat{h}^{(k)}$ is the maximum Hit-Rate that the k-th CP would obtain if it selfishly selected T_{col}, while $\hat{h}^{(k)}_{isp}$ is the Hit-Rate that it actually experiences according to the decision taken by the ISP. Notice that such hit-rates refer to the cumulative hit-rates of metro-aggregation and metro-core caches (i.e., it is the overall percentage of requests that the CPs serve from the area of the ISP).

In Tab. 4.4, we show the loss for each CPs of the first scenario described in the previous Section (5 CPs with an average number of contents of 5000 and contents' popularity distributed as shown in Fig. 4.9a). We notice that the loss highly varies among the CPs that, in this scenario, offer contents of significantly different popularities (e.g., CP_1's contents are much more popular, on average, than CP_5's contents). For instance, the loss goes from a minimum of $\sim 0.5\%$ to a maximum of $\sim 27.6\%$, which are experienced by CP_1 and CP_5 (the CP with the most and the least catalogues on average, respectively). In the considered scenario, there is a clear difference of popularity among the CPs. To understand the impact that popularity difference has on the loss, we perform additional simulations on a second scenario in which the contents' popularity is much more similar among the CPs. Also

Table 4.4: Loss of CPs' Hit-Rates when CPs offer contents with significantly-different popularity

CPs	$\mathbf{H_{isp}}$	$\mathbf{H_{cps}}$	**Loss**
$\mathbf{CP_1}$	0.807	0.811	0.499%
$\mathbf{CP_2}$	0.537	0.556	3.252%
$\mathbf{CP_3}$	0.147	0.171	14.104%
$\mathbf{CP_4}$	0.081	0.084	4.09%
$\mathbf{CP_5}$	0.068	0.094	27.64%

Table 4.5: Loss of CPs' Hit-Rates when CPs offer contents with similar popularity

CPs	$\mathbf{H_{isp}}$	$\mathbf{H_{cps}}$	**Loss**
$\mathbf{CP_1}$	0.659	0.67	1.65%
$\mathbf{CP_2}$	0.615	0.617	0.31%
$\mathbf{CP_3}$	0.596	0.596	0.03%
$\mathbf{CP_4}$	0.597	0.597	0%
$\mathbf{CP_5}$	0.533	0.533	0%

in this second scenario there are 5 CPs offering, on average, 5000 contents. The distribution of contents' popularity is derived setting $\sigma_1 = \sigma_2 = 15000$ and it is depicted in Fig. 4.9b. The loss of each CP in this second scenario is presented in Tab. 4.5, from which we can observe that the loss goes from a minimum of 0% to a maximum of 1.65% and it is therefore much less significant than in the previous case. From this comparison, it becomes evident that the difference in CPs' popularity highly affects the loss experienced by the CPs. This can be explained considering that the hit-ratios of the CPs do not significantly vary with changing the storage subdivision (as a result of tuning T_{col}) if the CPs cache contents with similar popularity. Hence, the hit-ratios of the single CPs do not strongly depend on T_{col} (i.e., the hit-ratios are similar and close to the optimum one regardless the T_{col} chosen by the ISP). We therefore conclude that the CPs are strongly penalized by being inhibited to select T_{col} only if their attractiveness towards the users is significantly different.

4.8.2.4 Complexity of the protocol and volume of the exchanged data

We now provide an evaluation of the data overhead introduced in all the phases of the execution of the protocol, as well as the time needed to perform them. The secure computation of the average size of CPs' contents' $\hat{s}$ is the only operation executed with our protocol that does not require the use of SSS. The k-th CP $1 \leq k \leq K$ sends to the RA the values of its number of contents $\mathcal{N}_k$ and the overall size of its catalogue $\mathcal{S}_k$ encrypted using the Paillier cryptosystem. Then, the RA decrypts these values and communicates to the ISP the ratio between them, i.e., $\hat{s} = \frac{\sum_{k=1}^{K} \mathcal{S}_k}{\sum_{k=1}^{K} \mathcal{N}_k}$. This operation requires the exchange of $2K$ messages between the CPs and the RA, and the exchange of one piece of data between the RA and the ISP. As all such data have negligible size (i.e., in the order of the hundreds of bits) and their computation is not time-consuming, the introduced time and data

Table 4.6: Overhead of data exchanged during the execution of the protocol (being ϕ the bit-length representation of the shares exchanged among the parties)

	ISP	RA	CPs
ISP	0	$\left(\frac{N!}{(N-2)!2!} + N\right) \cdot 9\phi^2$ $+N \log N \cdot 7\phi^2$ $+ 14\phi$	$4K\phi$
RA	$\left(\frac{N!}{(N-2)!2!} + N\right) \cdot 9\phi^2$ $+N \log N \cdot 7\phi^2$ $+N\phi + 14\phi$	0	$2K\phi$
CPs	$N\phi + 3K\phi$	$N\phi$	0

overhead can be considered negligible.

Let us now consider all the remaining operations, which are based on SSS. We refer to $\phi = \lfloor(\log_2 q + 1)\rfloor$ to indicate the bit-length representation of a share $\in \mathbb{Z}_q$. The secure computation of the multiplication triple $([\![a]\!], [\![b]\!], [\![c]\!]$ such that $c = a \cdot b)$ requires the ISP and the RA to exchange 4ϕ bit (to obtain $[\![a]\!], [\![b]\!]$ in a distributed manner) and other 6ϕ to obtain $[\![c]\!]$. The reader is referred to [37] for an in-depth understanding of all the required exchanges.

The collections of the shares relative to N requests issued during T_{col} requires the following exchange of data: $2N\phi$ (to account for the shares sent by the CPs to the ISP and the RA) and $N\phi$ for the shares of 1s sent from the RA to the ISP (to account for the value associated with each request). The next phase requires $N \log N$ equality tests to perform the aggregation of the collected shares and $\binom{N}{2}$ $+N$ comparison operations to compute the ranks of the contents and to compare them with the size of the cache. Each equality operation requires the exchange of $2\phi^2$ (which need to be exchanged during the execution of the protocol, i.e., *online*) and $12\phi^2$ (which can be pre-computed and transmitted before the execution of the protocol, i.e., *offline*) bits, while the comparison operation requires $18\phi^2$ bits exchanged on-line [112].

Then, the ISP and the RA compute the random value rv in a secure and distributed fashion and use it to obtain $[\![rv \cdot z_k \cdot S_c]\!]$ and $[\![rv \cdot \sum_{j=1}^{K} z_j]\!]$. This operations require the exchange of 18ϕ (2ϕ for the computation of rv and 16ϕ needed for the multiplications). Successively, the RA sends the obtained shares $[\![rv \cdot z_k \cdot S_c]\!]_{RA}$ and $[\![rv \cdot \sum_{j=1}^{K} z_j]\!]_{RA}$ to the K CPs, which requires $2K\phi$ additional transmitted bit. Finally, the ISP exchanges with the CPs their shares to obtain γ_k, and this results in an additional exchange of $4K\phi$ bits. In Table 4.6, we show the amount of data (in bits) exchanged by the three entities in a round of execution of the protocol.

From these considerations, it results that the additional time overhead T_{op} is given by the following formula:

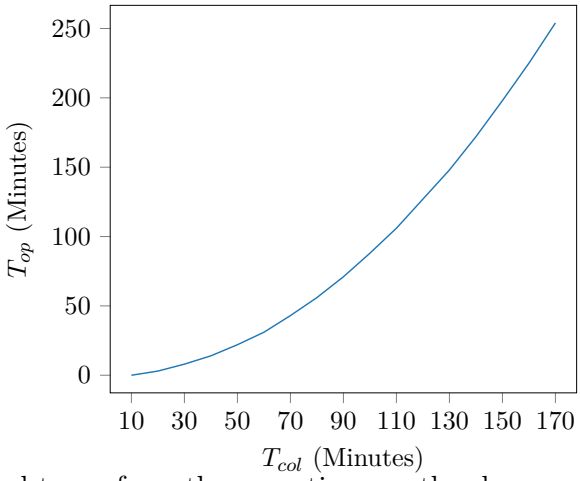


Figure 4.10: Time needed to perform the operations on the shares vs Period of collection of the shares (an arrival rate $\lambda = 1req/s$ is considered)

$$T_{op} = N \log N \cdot \tau_{eq} + \left(\frac{N!}{(N-2)!2!} + N \right) \cdot \tau_{comp} \tag{4.5}$$

Where τ_{eq} and τ_{comp} refer to the time required to perform an equality and a comparison operation, respectively. By discarding the operations that can be performed offline, we obtained $\tau_{eq} \simeq 0.47ms$ and $\tau_{comp} = 0.68ms$ on a Intel Core I7 computer. A representation of the time overhead needed to perform operations on the shares (i.e., T_{op}) as a function of T_{col} is depicted in Fig. 4.10.

Concerning the overhead introduced by the execution of the protocol, the volume of data exchanged with the CPs can be considered negligible. Conversely, the overhead of data exchanged between ISP and RA grows quadratically with the number of requests issued during the collection phase and with the bit-length representation of the shares (i.e., ϕ). With $\phi = 13$ bits, it is possible to generate unique shares during a collection phase that lasts up to 135 minutes, considering an arrival rate of $1req/s$. With these parameters, we obtain an overhead of $\simeq$ 2.2Gb online and $\simeq$ 5Mb offline considering $T_{col} = 80$ minutes. This overhead drops to $\simeq$ 1.2Gb when $T_{col} = 60$ minutes and to $\simeq$ 138Mb when $T_{col} = 20$ minutes. This overhead is acceptable, especially considering the traffic reduction achievable by the ISP. Remarkably, low values of T_{col} does not only guarantee the lowest data overhead, but also the lowest RO (as results from the analysis described in Section 4.8.2.2). Moreover, the negative impact of such overhead may be further reduced by colocating the RA with the ISP (e.g., as a virtual machine). We stress on the fact that the popularity-driven subdivision needs to be computed for each cache storage. In this work, we consider that the storage size can be of two types only: capacity of metro-core nodes and capacity of metro-aggregation nodes (i.e., 10% and 5% of the total size of the CPs' catalogues, respectively). Hence, the considered overhead needs to be accounted twice. Notice, however, that this overhead is still acceptable, and it would be acceptable even if more possibile caches' capacity was available. For example, if 10 types of

cache sizes were present, it would be required to execute the protocol 10 times. This would imply, considering $T_{col} = 60$ minutes, an overhead of 12 Gb, which is ~ 4 times the average size of the CPs' contents in our simulations.

4.9 Concluding Remarks

4.9.1 Summary

In this Chapter, we described a privacy-preserving network-neutrality-compliant protocol for caching of VoD contents in ISP networks. The protocol guarantees that the ISP assigns portions of its caches' storage to several CPs proportionally to the popularity of their contents (i.e., *popularity-driven* subdivision) and it is therefore compliant with neutrality requirements recently proposed in the literature. Besides ensuring a NN-compliant caching, the protocol also allows to meet CPs' and ISP's privacy requirements, as the information about contents' popularity and size of cache are not disclosed. We evaluated how caching performance is influenced by a popularity-driven-subdivision in terms of overall network resource occupation and hit-ratio for ISP and CPs comparing it to baseline approaches, namely, *static subdivision*, where CPs are assigned the same amount of storage independent of their popularity, and *resource-occupation-driven*, where CPs are assigned an amount of storage according to amount of capacity their requests occupy in ISP's network. We performed two main sets of simulations. In the first one we considered a simplified version of the protocol and we evaluated its effectiveness against the static subdivision only, which was significantly outperformed in terms of Hit-Ratio. These simulations also allowed us to show the scalability of the protocol with increasing number of CPs and number of their contents. We also developed a dynamic VoD content caching and distribution simulator to perform more extensive simulations considering all the aforementioned baselines, the complete protocol and a more complex network of cache servers. We found that the popularity-driven and the resource-occupation-driven subdivisions lead to a reduction of the RO of up to $\sim 52\%$ (and to an improvement of the hit-ratio of up to $\sim 32\%$) with respect to the static subdivision. In particular, the minimum RO and the maximum hit-ratio are obtained with the popularity-driven subdivision computed with our protocol. Moreover, we observed that the RO is highly-influenced by the frequency of execution of the protocol, that we assume to be tuned by the ISP in order to minimize the RO. Numerical results show that each CP experiences a loss in terms of its hit-ratio with respect to the case where it could selflishly establish this frequency. In the considered scenarios, this loss can range from a minimum of 0% to a maximum of $\sim 27\%$ and it is much less significant when CPs' popularity are similar. Overall, our protocol proved to be beneficial in increasing caching performance (e.g., RO is reduced) while ensuring the protection of privacy. Note that privacy is protected also using the benchmark approaches, but none of them guarantees that the subdivision is actually compliant with NN requirements (as our protocol, instead, ensures). We also evaluated the data overhead introduced by the protocol and we conclude that it is

acceptable compared to the reduction of RO experienced by the ISP. As a future work, we plan to extend our study considering more challanging security models (e.g., malicious parties that can alter their data during the execution of the protocol).

4.9.2 Final Comments

In this Chapter we have considered an advanced video content delivery approach enabled by the application of caching strategies. Specifically, we have considered a cooperative approach between an ISP and several CPs that remotely manage the caching resources. The protocol that we have proposed allows to compute a subdivision of the available caching resources that is both compliant with Network Neutrality ideals and privacy preserving, as ISPs and CPs are not required to exchange with each other confidential information. In the next Chapter, we consider a similar content delivery model enabled by the use of Virtual Servers that the CP uses to stream Live Videos directly from the area of the ISP. Specifically, we employ an existing privacy-preserving data sharing protocol that allows the ISP and the CP to deploy such servers close to users' requests without exchanging sensitive information (i.e., the CP does not discover users' location and the ISP does not discover users' requests).

Protection of the Privacy of Live Videos' Users' Location 5

As described in the previous Chapters, VoD content delivery benefits from the application of caching strategies implemented by the ISPs. Differently from traditional VoD contents, Live Videos (LVs) are delivered (i.e., streamed) while they are being generated. Hence, LVs cannot be stored in advance in cache servers. However, the LVs can be directly streamed from Virtual Server (VSs) that are deployed in strategic ISPs' network positions to achieve some optimization objective (e.g., minimize the average retrieval latency). The knowledge of the geographic distribution of users' requests is crucial to perform an optimal VSs deployment, mostly because the requests for LVs are characterized by higher locality with respect to traditional VoD [100]). Due to contents' encryption, this geographic distribution is unknown to the ISP. In this Chapter, we employ an existing protocol to enable an ISP discover the geographic distribution of the requests in a privacy-preserving manner. Based on this information, the ISP deploys the VSs within its network. We show that primary privacy requirements can be achieved at no expenses of QoE, which is evaluated considering the average number of network hops crossed by the LVs to reach their viewers. We also show that more stringent privacy requirements can only be met if the CP applies data perturbation strategies that induce a trade-off between QoE and users' privacy, that we also evaluate by means of simulations.

5.1 Motivation

A multimedia service that is rapidly gaining popularity is LV, which is expected to account for the 13% of global Internet video traffic by 2021 [91]. Several Content Providers (CPs) offer platforms that allow to stream LVs, e.g., Facebook, YouTube and Twitch. The most common approach implemented by CPs consists in streaming the LV directly from their data centers. As such servers are generally located far away from the final users, this solution may lead to a poor QoE experienced by users.

Thanks to the use of virtualization strategies, the CP can improve the QoE by streaming the LVs from Virtual Servers (VSs) (i.e., Virtual Machines that receive the LVs from the CP and stream them to the users) hosted within the network of the ISP.

The requests for LVs are characterized by higher locality with respect to traditional VoD contents [100]. This is explicable considering that LVs are often user-generated contents that become viral within small geographical areas (e.g., a school or a village). Therefore, an efficient deployment of the VSs requires the ISP to know the geographic distribution of users' requests. Since the ISP delivers traffic to the users located within its area, the ISP also knows their position but, due to content encryption, it is not aware of the contents they request. Conversely, the CP has full knowledge of what the costumers request, but it inaccurately infers their position. These information are complementary and, if exchanged between the two parties, would enable the ISP to know how requests are geographically distributed and perform the optimal VSs deployment accordingly. Conversely, only a suboptimal VSs deployment can be executed.

In a privacy-preserving scenario, however, this exchange should be avoided. Specifically, to guarantee users' privacy, the CP is not allowed to know precisely the location of its users, whereas the ISP should not know that contents that they request. In this work, we envision a content delivery system in which the ISP and the CP cooperate towards the privacy-preserving discovery of the geographic distribution of users' requests. To this end, we employ an existing *Secure Multiparty Computation* protocol that the ISP and the CP can apply to obtain the geographic distribution of costumers' requests in a privacy-preserving manner. By using this protocol, which is described in Section 5.3.2, the ISP obtains an aggregate information about users' geographic distribution, which is then used to optimally deploy the VSs within its network (task performed by means of an algorithm presented in Section 5.3.1).

To make the cooperation between ISP and CP maximally beneficial to both the parties, we assume that the ISP deploys the VSs to minimize the average number of hops crossed by the LVs to reach their viewers. The minimization of this metric, in fact, allows the CP to offer an increased QoE (as contents' retrieval latency and congestion probability are lowered) and the ISP to reduce its network resource occupation (as traffic is delivered from sources close to users). We evaluate the average number of crossed network hops if the information about the geographic distribution of LVs' requests is known to the ISP, and we compare it with that obtained if, instead, an inaccurate request distribution is estimated without executing the privacy-preserving protocol. Obtained results show that the application of the protocol allows to achieve a significant reduction of the number of crossed network hops.

We define the privacy requirements of the ISP and the CP and we evaluate their fulfilment when the privacy-preserving protocol is employed. In particular, the privacy of the ISP is fulfilled when the CP is not able to infer, from the execution of the protocol, additional information about the location of the final users. On the other hand, the privacy of the CP is guaranteed when the ISP cannot obtain additional information about users' requests (namely, the contents of users' requests and if

two users are requesting the same content). We formally describe these requirements in Section 5.4.1.

We notice that ISP's privacy is guaranteed at no expenses of the effectiveness of contents' delivery (measured considering the average number of network hops crossed by the LVs to reach the users). Instead, a complete fulfilment of CP's privacy requirements can only be achieved if the CP alters its data during the execution of the protocol (by employing data perturbation strategies). The level of this perturbation induces a trade-off between privacy and effectiveness of the delivery. Specifically, a high level of CP's privacy comes at the cost of a significant increase of the number of crossed network hops.

5.2 Related Work

Content delivery is a complex ecosystem composed of several entities (e.g., CPs, ISPs and CDNs), none of which has the end-to-end control of the transmitted data [60]. Cooperation among the involved players (CPs and ISPs, in particular) is regarded as a natural direction to address the performance issues resulting from the fact that contents traverse different domains (i.e., the networks of the CP, CDN and ISP) [52]. Several cooperative strategies have been proposed and analyzed to improve participants' QoS [52], showing that virtualization is an enabling technology for effective cooperation [60]. The benefits of virtualization of computational resources, in fact, are universally recognized (e.g., an increased scalability and a more flexible adaptation to traffic dynamics [101]). However, the effective deployment of virtual resources in a multi-domain environment may require the involved entities to share information that violate users' privacy (e.g., their location and the resources they access, which is the specific problem that we address in this Chapter).

Althougth the literature concerning the optimal deployment of virtual resources is particularly rich and heterogeneous (e.g., energy-efficient [76] and network-resource-efficient [13] migration of virtual machines), the problem of efficiently deploying virtual resources for streaming live events has attracted attention only recently. To our knowledge, the problem has been addressed for the first time in [80], where an ISP's network is the considered domain. The main shortcoming of this work is that the geographic distribution of the requests is assumed to be known to the ISP, while under content encryption it is not. Our work solves this problem by allowing an ISP and a CP to infer such distribution by employing an existing privacy-preserving protocol. This protocol falls in the category of Secure Multiparty Computation methods, which allow multiple parties to jointly compute an optimization function while keeping their inputs private. These methods find wide application, e.g., in computer security and financial data analysis [73].

5.3 Building Blocks of the Cooperation

We now present the two main building blocks of the cooperative architecture, namely an algorithm for efficiently deploying the VSs and a protocol to allow the private computation of the cardinality of the intersection of two sets.

5.3.1 Virtual Server Placement Algorithm

Given a catalogue of LVs, a geographic distribution of users' requests and a set of *Candidate Nodes* (CNs) belonging to the ISP network, the ISP performs the deployment of the VSs relative to the live videos on the CNs in order to achieve some optimization objective. The authors of [80] propose both exact and heuristics solutions to carry out this operation with the objective of minimizing the global latency experienced by viewers. Due to the expensive computational requirements of the exact solution, in our simulations we adopt the heuristic solution that we briefly describe here. A fitness value is defined to quantify the goodness of placing a specific VS content on a CN. Specifically, the fitness value takes into account i) the total latency experienced by the users and ii) the overall traffic introduced by a content when it is streamed from a specific CN. Constraints are defined for the capacity of links, while the storage of the hosting nodes is considered unlimited and only one VS for each content is instantiated in the network. Among the potential solutions that fulfill the constraints, the one with the highest fitness value is chosen.

5.3.2 Privacy-preserving protocol

The problem of privately inferring the geographic distribution of LV contents requests can be solved by employing an existing protocol presented in [38]. The protocol allows to efficiently and privately compute the cardinality of the intersection of two sets (owned by separate entities that do not will the reveal the other party their private information). One of the two entities plays the role of *client*, while the other is the *server*. After the application of the protocol, the client learns the cardinality of the intersection of the two sets. In addition, both the client and the server learn an upper bound of the cardinality of the other party's set. The overhead introduced by the protocol is linear in the cardinality of the input sets.

5.4 Problem Statement

ISP and CP are assumed to be separate entities with different domains. The domain of the ISP is represented by a set $CN = \{CN_1, CN_2, ..., CN_K\}$ of candidate nodes that host the live contents of the catalogue C owned by the CP (i.e., $C = \{C_1, C_2, ..., C_M\}$). The area covered by the ISP can be represented as a set of non-overlapping areas, i.e., $A = \{A_1, A_2, ..., A_N\}$. The domain of the CP is abstracted as a *Video Server* external to the area of the ISP.

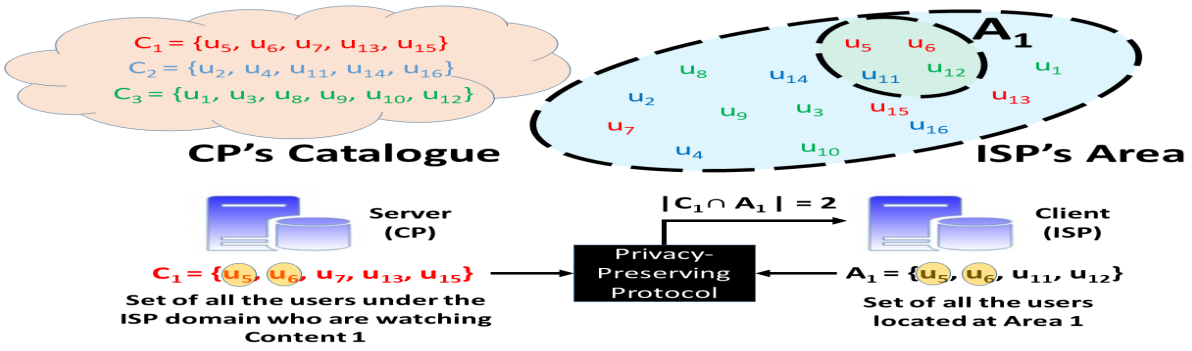


FIGURE 5.1: Example of privacy-preserving exchange of users' information

Each user is located in an area A_j, from which she issues her requests to the CP. Both the ISP and the CP can recognize a user from her identifier $u_r, \forall r \in \{1, 2, ..., R\}$ (e.g., the IP address). For the sake of simplicity, we assume that each user issues a single request. As the ISP is in charge of routing the contents to the users belonging to its domain, we assume that it can associate each user with the area A_j from which it streams the live content.

Both the CP and the ISP aim to efficiently place the live contents closest to the users, as this results in an improved QoS and in the reduction of traffic. More formally, they aim at finding the best Candidate Node k (CN_k) from where the i-th content (C_i) should be streamed in order to minimize the average number of hops (h) crossed by the live contents to reach the viewers:

$$\min \sum_{j=1}^{N} \sum_{i=1}^{M} \sum_{k=1}^{K} \mathcal{D}(i,j) \cdot x_{ik} \cdot h_{ijk} \tag{5.1}$$

where $\mathcal{D}$ is a matrix representing the geographic distribution of users' requests, i.e., its (i,j)-th entry is the number of requests for content C_i issued from area A_j; $x_{ik} = 1$ if C_i is hosted in CN_k (0 otherwise) and h_{ijk} is the number of hops crossed by content C_i to reach area A_j when it is hosted in CN_k. Following [80], capacity constraints are defined for the network links but not for the CNs, as detailed in Section 5.6.

The algorithm employed to perform such optimization, referred to as Γ, takes as input $\mathcal{D}$ and other system parameters (e.g., network settings) and returns the best CN_k for every content C_i. Note that the approach described throughout this Chapter can be applied to other metrics and objective functions (e.g., the maximum load on network links). We assume that algorithm Γ is executed by the ISP, because the CNs belong to its domain. However, the ISP lacks the knowledge of the input $\mathcal{D}$, which prevents the application of the optimization algorithm. In fact, the ISP only knows the geographic distribution of its customers, while the CP knows what its customers request but inaccurately infer their geographic distribution. In Section 5.4.1, we describe the privacy requirements that prevent the two parties from freely exchanging these information about the

distribution of requests. We then present in Section 5.5 a protocol for making CP and ISP jointly compute $\mathcal{D}$ without revealing the pieces of information that they own.

5.4.1 Privacy Requirements of ISP and CP

The CP requires that the ISP does not discover i) the content that is being streamed by the generic user u and ii) if two users u_1 and u_2 are streaming the same content. We refer to the first requirement as *primary* privacy CP's goal and to the second as *secondary* privacy CP's goal. The importance to guarantee the primary requirement is widely-recognized in the literature (e.g., [116]), as it prescribes the protection of users' requests confidentiality. To the extent of our knowledge, we are the first to consider the privacy requirement embodied in the secondary goal. However, we argue that such privacy objective is still very relevant, as it prevents the ISP from being able to profile theirs users (i.e., by grouping together those with similar interests).

On the other hand, the ISP requires that the CP cannot associate the generic user u with the area in which it is located. Similarly to the primary ISP's privacy requirements, there is wide agreement on the importance of protecting users' location [22, 18].

5.5 The Privacy-Preserving Cooperative Protocol

For every area $A_j \in A$, the ISP builds a set $\mathcal{A}_j$, whose elements are the identifiers of all the users located in the j-th area, regardless of the content they are watching. Similarly, for every content C_i, the CP builds a set $\mathcal{C}_i$ containing the identifiers of the users that are currently requesting the i-th content. The identifiers could be, for instance, the IP addresses of the users, or any attribute that allows both the ISP and the CP to unequivocally identify the requests.

$\mathcal{D}(i,j)$ is the number of requests for content C_i issued from the generic area A_j, i.e., the **cardinality of the intersection of sets** $\mathcal{C}_i$ and $\mathcal{A}_j$.

$$\mathcal{D}(i,j) = |\mathcal{C}_i \cap \mathcal{A}_j| \tag{5.2}$$

Due to the privacy requirements presented in Section 5.4.1, however, the two parties cannot reveal the contents of their sets, which prevents them from computing a simple intersection. This problem can be solved by employing the secure multyparty computation protocol described in Section 5.3.2. More specifically, the CP is the server and the ISP is the client that learns $\mathcal{D}$, that it successively uses as input of the algorithm Γ. An example of application of the protocol is depicted in Fig. 5.1, where it is shown how the ISP infers the number of users in area A_1 who are watching content C_1.

5.5.1 Attackers' Model

We assume that both CP and ISP honestly execute the protocol, but perform analysis on the obtained data to infer additional information about the other party's inputs (i.e., *honest-but-curious*

attacker model).

As far as the ISP privacy is concerned, the CP knows the identifiers of the users $u_r, \forall r \in \{1, 2, ..., R\}$ (e.g., it knows their IP addresses in order to deliver the contents to them). As server of the privacy-preserving protocol, the CP also obtains an upper bound of $|A_j|, \forall j \in \{1, 2, ..., N\}$, from which it tries to associate the generic user u with the area where it is located. We assume that the CP can localize its users by using inaccurate techniques (e.g., IP geolocation) that can mis-locate users over the whole area of the ISP. This is confirmed in [98], where it is shown that the localization error can be significant (i.e., up to hundreds of kilometers). $|A_j|$ provides the CP with the information about the number of users located within the j-th area, but not which contents they request. Hence, the privacy requirement of the ISP is fulfilled.

As far as the CP privacy is concerned, the ISP can perform its analysis on the following pieces of information: identifiers of the users $u_r, \forall r \in \{1, 2, ..., R\}$ and geographic distribution of the users $\mathcal{A}_j, \forall j \in \{1, 2, ..., N\}$ (owned because needed to perform the delivery of contents). In addition, as client of the privacy-preserving protocol, the ISP learns the geographic distribution of the number of requests $\mathcal{D}(i, j), \forall i \in \{1, 2, ..., M\}$ and $\forall j \in \{1, 2, ..., N\}$ and an upper bound of total number of users requesting a particular content $|C_i|, \forall i \in \{1, 2, ..., M\}$. From these information, the ISP tries to infer i) if a content C_i is streamed by the generic user u_r (*primary* privacy objective) and ii) if two users u_1 and u_2 are watching the same content (*secondary* privacy objective).

ISP and CP execute the privacy-preserving protocol $N \cdot M$ times, i.e., once for each pair of $\mathcal{A}_j$ and $\mathcal{C}_i$. In order to build $\mathcal{D}$ the ISP needs to unequivocally identify to which content the current execution of the protocol refers. This objective can be achieved by associating each content with a pseudonym that is freely readable by the ISP, thus allowing the CP to not reveal the actual names of contents. From $\mathcal{D}$, the ISP only discovers an aggregate information on the requests for (the pseudonym of) C_i coming from A_j, while from A_j the ISP knows the identifiers of the users located in the j-th area. From this information, the ISP can infer the probability that a generic user u located in A_j is requesting C_i as:

$$P_u(C_i) = \frac{\mathcal{D}(i, j)}{|\mathcal{A}_j|} \tag{5.3}$$

Hence, the ISP can make a guess on the pseudonym that a user u is requesting, but it can never link it to the corresponding actual content name, thus fulfilling the *primary* requirement.

From Eq. (5.3), it is possible to mathematically derive the probability that two generic users u_1 and u_2 are simultaneously watching content C_i:

$$P_{u_1 u_2}(C_i) = P_{u_2}(C_i | u_1 \text{ watches } C_i) \cdot P_{u_1}(C_i) \tag{5.4}$$

and, from this, the probability that u_1 and u_2 are watching the same content is:

$$P_{u_1u_2} = \sum_{i=1}^{M} P_{u_1u_2}(C_i) \tag{5.5}$$

Hence, the *secondary* privacy requirement is fulfilled to a degree that is measured as $1 - P_{u_1u_2}$, i.e., the probability that the ISP cannot recognize if two users are streaming the same content.

5.5.2 Countermeasure Description

The ISP is capable of violating the *secondary* privacy requirement with a certain probability. As a countermeasure, the CP can execute the protocol by applying a perturbation to its data as follows. Given a content C_i, the CP builds the set $\mathcal{C}_i$ as explained in Section 5.5. The noisy counterpart of $\mathcal{C}_i$, referred to as $\tilde{\mathcal{C}}i$, is obtained by replacing, with a given probability p, each user u_x who is currently streaming C_i (i.e., $u_x \in \mathcal{C}_i$) with another user $u_y \notin \mathcal{C}_i$ (i.e., who is streaming another content). This leads to the joint computation of a noisy version of $\mathcal{D}$, that we refer to as $\tilde{\mathcal{D}}$, that improves users' privacy at the expense of a performance degradation that we quantitatively assess in Section 5.7.2.

The effect of this perturbation on the probability that two generic users u_1 and u_2 are streaming the same content is given by Eq. (5.6), which is derived considering that both the users actually belong to $\mathcal{C}_i$ if and only if they had not been moved from another set $\mathcal{C}_w \neq \mathcal{C}_i$, which happens with probability $(1-p)^2$:

$$\tilde{P}_{u_1u_2} = (1-p)^2 \cdot P_{u_1u_2} \tag{5.6}$$

where $\tilde{P}_{u_1u_2}$ is the probability that the ISP discovers that two generic users are watching the same content if the CP applies a perturbation to its inputs. In this case, the degree of fulfilment of the secondary privacy requirements is measured as $1 - \tilde{P}_{u_1u_2}$.

5.6 Simulation Settings

5.6.1 Traffic Modeling

The start time of a live-video stream is chosen according to a uniform probability distribution, whereas its lifespan is distributed according to Poisson distribution with mean value of 24 minutes. The duration of a content is not assumed to be related to its popularity. Users watch a content for a random period that lasts, on average, half of its lifespan. A Zipf distribution with skewness parameter $\alpha = 0.8$ models the popularity of the 80 contents of the catalogue (the size of the catalogue is derived from [80]).

Content C_i can be requested from $\beta \cdot \frac{M-i}{M} \cdot |A|$ number of areas (which are considered adjacent to model the spatial correlation of requests). M is the total number of contents, $|A|$ is the total number of areas and i is the position of the content in the popularity rank. According to this model, the

least popular contents are requested from a fewer number of areas (i.e., they have a higher degree of spatial locality). $\beta \in [0, 1]$ tunes the degree of spatial locality of the catalogue. Specifically, low values of β make most of the contents to be requested from a small cluster of adjacent areas.

5.6.2 Algorithm for deploying the live-video contents

We slightly modify the heuristic solution proposed in [80] in order to minimize the average number of crossed hops (instead of the latency) between viewers and candidate nodes. In fact, this metric is more objective than the latency, which is heavily biased by the arbitrary simulation settings. Moreover, we consider a different topology than that described in [80], which does not represent well the hierarchical structure of today's ISPs' networks.

5.6.3 Network Settings

We consider a three-tier metro network that covers a squared area of 100 km^2, which approximates the area of an average-size city. We consider this assumption to be realistic, as most of today's telecom networks are arranged in such hierarchical fashion. The first layer is composed of 400 BSs located at equidistant positions. We assume that the ISP can localize its users up to the BS they are connected to, i.e., the j-th BS specifically covers area A_j. Each BS is connected to the nearest node among the 25 available Access Nodes (ANs), which are uniformly distributed over the whole area. The metro segment is composed of 5 Metro Nodes (MNs) interconnected in a ring fashion. Each AN is connected to its nearest MN, which in turn has a dedicated link toward a remote server that abstracts the CP. A representation of the employed topology is drawn in Fig. 5.2. Both ANs and MNs can host and stream the live videos (i.e., they are candidate nodes). Following [80], no constraint is defined for the storage of the CNs. Concerning the capacities of network links, we reserve 50 Mbps for the transmission of the live video in the backhaul links (from ANs to BSs). Because the MNs are missing in the original topology, we arbitrarily set the available capacities to 100 Mbps and 200 Mbps for the links between MNs and ANs and for the links among MNs, respectively.

5.7 Results

Ten requests are issued every minute for a total period of 120 minutes, according to the traffic model described in Section 5.6.1. The privacy-preserving protocol to obtain the request distribution $\mathcal{D}$ and the algorithm for deploying the VSs (i.e., Γ) are executed every fixed period T. We run 200 simulations and we show the average of the performance.

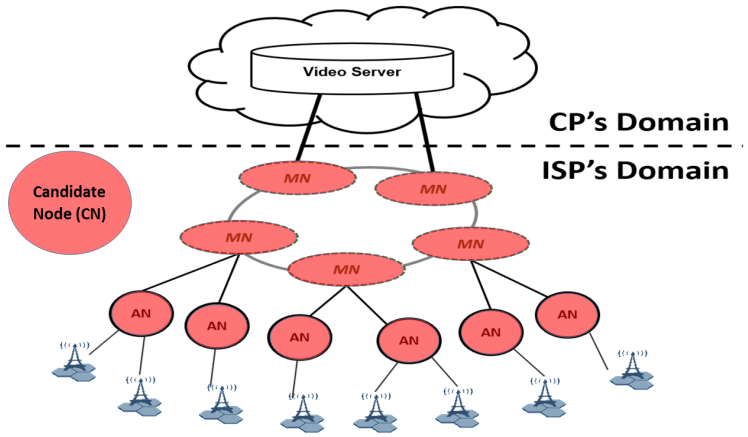
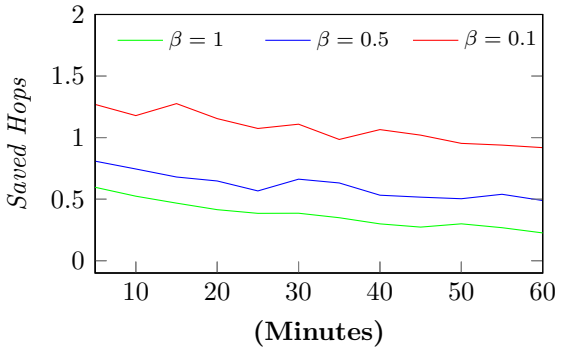
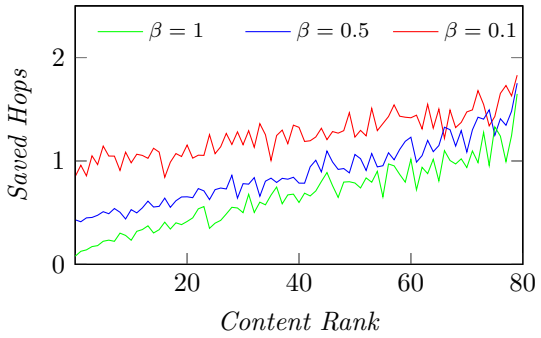


Figure 5.2: Considered topology



(a) Assessment of the gain for varying periods T between two successive executions of Γ



(b) Performance Gain obtained when algorithm Γ is fed with $\mathcal{D}$ instead of $\mathcal{D}_{mis}$

Figure 5.3: Privacy of the most popular contents of the CP's catalogue considering a privacy threshold $\eta = 0$

5.7.1 Impact of the mis-location

Firstly, we evaluate the average number of hops saved by feeding Γ with the input $\mathcal{D}$ instead of its mis-located counterpart $\mathcal{D}_{mis}$ that is obtained as follows: the CP infers the location of its users by employing geo-location techniques (that are assumed to mislocate users over the all area covered by the ISP) and sends it to the ISP. We show the performance for $\beta \in \{0.1, 0.5, 1\}$ with varying T in Fig. 5.3a. The maximum gain is around 1.28 hops (obtained with $T = 15$ minutes and $\beta = 0.1$) and can be considered significant in the examined topology, where, given a fixed routing based on the shortest paths, the farthest candidate nodes are 4 hops away from each other. As expected, the gain generally increases with decreasing T and is higher for low values of β.

On one hand, low values of T implies a frequent re-deployment of the VSs, which allows to adapt

to the dynamics of the requests. Note that every time Γ is executed, the ISP has to obtain the input $\mathcal{D}$ by engaging an exchange of data with the CP by using the privacy-preserving protocol, which introduces the following traffic overhead between the CP and the ISP [38]:

$$Overhead = \sum_{j=1}^{N} \sum_{i=1}^{M} \Big(2 \cdot (|\mathcal{A}j| + 1) \cdot q + |\mathcal{C}i| \cdot v \Big) \tag{5.7}$$

where q and v are the size (in MB) of two security parameters. To understand if this overhead is significant, we computed it for all the pairs of $\mathcal{A}j$ and $\mathcal{C}i$ in our simulations and we obtained an average value of around 11.65 MB per simulation (i.e., around 0.15 MB per live video) considering q and v to be 2048 bits. Only the value of q was suggested in [38], while we agreed on the value of v based on considerations about how it affects the security of the protocol. Hence, the overhead can be considered negligible also if Γ is executed frequently.

On the other hand, the gain increases with decreasing β, because the mis-location error is more detrimental when the catalogue is characterized by high locality. Similarly, the gain is more significant for the least popular contents that, according to our model, have a higher degree of locality. We depict in Fig. 5.3b the results obtained with $T = 30$, where the gain is shown for each content in decreasing order of popularity. The average gain for the least popular content is around 1.75 over a maximum of 4 hops. The effects of mis-location are almost negligible on highly-popular contents when $\beta = 1$, but are quite significant for the least popular ones. Conversely, decreasing β has the effect of increasing the degree of locality of all the contents, thus making the gain relevant for all the catalogue. The gain difference between least and most popular contents is reduced but remains significant. Hence, the application of the privacy-preserving protocol benefits the contents to a degree proportional to their locality.

5.7.2 Impact of input perturbation

In Fig.5.4 we evaluate the trade-off between CP's privacy and performance for $T = 30$ and $\beta \in \{0.1, 0.5, 1\}$. The probability p tunes the level of distortion of the input and rules this trade-off. Privacy is measured as the average probability that the ISP cannot infer if two users are watching the same content (i.e., secondary privacy requirement) and it is computed using Eq. (5.6) for all the pairs of users in our simulations. We quantify the performance as the average number of hops lost when Γ is fed with the noisy request distribution $\tilde{\mathcal{D}}$ instead of its actual counterpart $\mathcal{D}$.

We notice a general decrease of the performance with increasing p, that is more significant for low values of β. In the latter case, in fact, a relevant number of contents is requested from small areas, thus increasing the probability to correctly guess if two co-located users are watching the same video. If $p = 0$ (i.e., no perturbation), the privacy of users is around 0.15, 0.17 and 0.2 for $\beta = 1, 0.5$ and 0.1, respectively. A very high level of privacy (i.e., close to 1) is achieved at the cost of around 0.3,0.6 and 1 lost hops which, compared with the performance gain shown in Fig. 5.3a

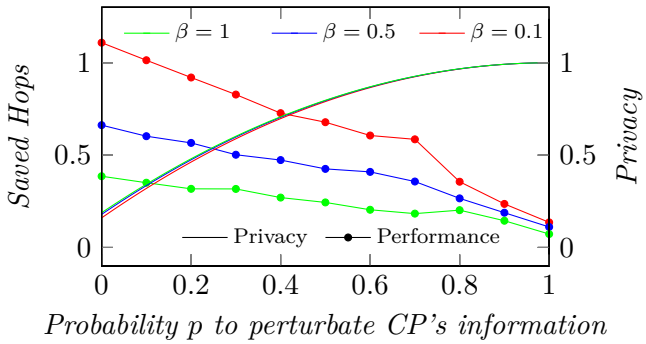


Figure 5.4: Performance/Privacy Trade-off

for $T = 30$, results in a net performance gain of 0.09, 0.06 and 0.1 average hops, for $\beta = 1, 0.5$ and 0.1, respectively. Hence, a complete achievement of the secondary privacy requirement significantly impairs the performance. Conversely, the fulfilment of the primary privacy constraint, which is much more relevant, comes at no performance loss.

We remind that the described results are obtained considering the minimization of the total number of network hops crossed by the users to retrieve the live videos. As mentioned in this Chapter, this metric allows us to perform an evaluation of retrieval performance that is less dependent on the simulation settings (i.e., the delay introduced by each network link). However, the CP and the users are likely to be more concerned with the minimization of the retrieval latency, as it provides a more accurate indication of the QoE. We stress that the considered optimality criterion (i.e., minimization of the number of crossed hops) coincides with the minimization of the retrieval latency only if all the network links introduce the same delay. As this assumption hardly holds in practical cases (i.e., in real telecom network), we plan, as a future work, to perform a thorough evaluation of the impact that links' delays have on the optimization process and, in particular, on the gain given by the use of our privacy-preserving protocol.

5.8 Concluding Remarks

5.8.1 Summary

In this chapter we applied an existing secure multiparty computation protocol to allow an ISP and a CP to jointly compute the geographical distribution of the amount of requests issued toward a CP from the ISP's area. In order to execute the protocol, the two parties do not need to reveal sensitive information, namely what users request (CP's privacy) and where users are (ISP's privacy). As a use case, we considered the deployment of Virtual Servers to stream live-video contents owned by the CP directly inside the ISP's network. By employing an existing algorithm to place the VSs, we concluded that the knowledge of the actual geographical distribution of requests leads to a significant

performance gain, which confirms the need to apply the privacy-preserving protocol to infer it.

We then proposed a strategy to violate the privacy requirements of both CP and ISP and a countermeasure to address it. Specifically, we assessed the trade-off between privacy and performance when the CP applies a perturbation to its information during the execution of the protocol. We realized that a basic privacy requirement (i.e., not allowing the ISP to discover which content is being streamed by a generic user) comes at no performance cost. Conversely, more stringent privacy requirements lead to a relevant performance degradation. Finally, the proposed cooperative approach is not limited to the considered use case, but it can be reproduced in many scenarios where it is required to move resources to the edge of the network (e.g., computational resources) in a privacy-preserving fashion.

5.8.2 Final Comments

In the work described in this chapter we have proposed an approach to guarantee both users' location privacy and service effectiveness in the context of live-video content delivery. The protocol proposed to achieve this objective is executed by the CP and the ISP and does not require any intervention of the final user. In other scenarios, instead, the user is in direct control of the information of her location and, therefore, is also expected to play a more relevant role in protecting it. For example, in Online Social Network platforms (e.g., Twitter) users can deliberately expose their location along with the contents they publish. In these cases, it is of paramount importance that users become aware of the risks associated with such public exposition of their sensitive data and understand how to protect it. In the next Chapter, we consider the problem of protecting the location of Twitter's users. Specifically, we measure the ability to infer unexposed locations from publicly-available ones. Then, we also propose approaches that users can employ to control their level of privacy and to quantitivaly measure the impact of the factors affecting it (e.g., the number of published locations).

Measurement and Control of Geo-Location Privacy on Twitter 6

In the previous Chapter, we have considered the problem of providing efficient live video content delivery while protecting users' location privacy. Location, in fact, is an information that most of the users tend to consider sensitive. At the same time, users' location is a business-relevant asset that many entities may be interested to obtain, e.g., a Location Based Service (LBS). Moreover, as Online Social Network (OSN) platforms allow users to publish their contents along with their location (i.e., in the form of *geo-tags*), this information is often publicly-available. All these facts call for the development of methods to increase users' awareness about location privacy, where by awareness we refer to their ability to quantify their level of privacy and to properly control the factors affecting it. In this Chapter, we consider the problem of protecting Twitter users' locations. In particular, we show that users' location can be accurately estimated based only on publicly available locations shared by users on this OSN platform. Therefore, we propose data perturbation techniques that users can enforce to control the exposition of their data, and we show the resulting privacy improvements. To shed light on the factors influencing users' vulnerability, we model privacy as a combination of users' social and behavioral characteristics. Finally, to further increase users' awareness of the effects of applying privacy control strategies, we examine, as a study case, the trade-off between users' privacy and the effectiveness of a proximity marketing LBS.

6.1 Introduction

OSNs have gained tremendous popularity worldwide. Just as an example, as of April 2019, Twitter and Facebook count around 330 and 2375 millions active users, respectively[1]. In OSNs users can perform several activities, such as socialize and publish different contents (e.g., opinions, news,

[1] https://www.statista.com/

videos, etc...). By doing so, users leave a *digital shadow* that, if properly analyzed, can provide very detailed information about them. Such information can be exploited by a third party to offer increasingly-tailored services to the users, e.g., a marketing company that uses social media platforms to perform targeted advertisement.

Users defend their privacy by limiting the amount of contents they share. However, this approach does not totally protect users from the disclosure of their personal data, since it has been proven that sensitive information about a given person can still be obtained from data released by other users [77, 16]. In fact, users more likely interact with and connect to people similar to them [86, 50] (e.g., with same interests, who visited common locations, etc...). For instance, a user can be mentioned by others in relation to a given topic, which may unintentionally reveal interest for that subject. Hence, a user is not in full control of the public exposure of her personal information [53], nor she can easily measure the related privacy risks.

In this respect, geographical location is widely considered both a highly-valuable and a very sensitive information. For example, users' position is often required for effective delivery of Location-Based Services (LBSs) [67], e.g., proximity marketing, in which users are advertised offers from retailers close to them. However, to protect their privacy, users rarely reveal their location in OSNs (operation referred to as *geo-tagging*). As an example, only less than 1% of the messages published on Twitter (i.e., the *tweets*) are provided with a geo-tag [55]. Nonetheless, third parties that offer a LBS might have alternative strategies to obtain users' location[2]. In previous studies, it has been proven that users' unexposed location, even if not explicitly exposed, can still be accurately inferred by combining different sources of information, such as users' generated contents (e.g., public messages), mobility patterns and social cues [45, 103, 107].

In this study, we explore the problem of how to measure and control a user's geo-location privacy on OSNs. We consider a scenario where a third-party entity (e.g., a LBS) is interested to obtain users' location. This third party is the attacker that tries to violate users' privacy by inferring the locations that users are not willing to expose. In such scenario, our first objective is to measure the level of geo-location privacy of a user. We define the *geo-location privacy* of a user as the geographical distance between the actual location and the one estimated by the attacker. More specifically, we investigate whether a user's privacy can be violated by leveraging the locations shared by other users. Notice that such attack does not require the implementation of any illicit strategies, as it is based only on the information attainable from publicly-available geo-tags.

As users are generally not aware of the potential privacy risks behind this public data exposure, it is crucial to provide them with tools to measure the geo-location privacy and control it (i.e., to be capable of setting the level of privacy a user is comfortable with). In this work, we consider the Twitter OSN as a study case and we provide the following four main contributions:

- To assess users' geo-location privacy, we propose a novel deep learning architecture that,

[2]The terms geo-tag and location are used interchangeably in this study.

starting from publicly-available geo-tags, attempts to violate users' privacy by unveiling their unexposed locations. Our results confirm the serious concerns about location privacy in OSN, as on a test dataset, we show that the geo-location privacy of about 60% of the analyzed users can be violated with a precision below 1km.

- We propose two data perturbation techniques that users can employ to control the public exposure of their geo-tags and, in this way, improve privacy. These strategies are called geo-tag obfuscation and geo-tag reduction and, by using them, a user can provide noisy locations or diminish the number of shared geo-tags, respectively. We found that the former approach is preferable to the latter as it allows a more effective tuning of the desired privacy level.

- We model the privacy value obtained with the deep learning approach as a function of users' characteristics related to mobility, social network and enforced data perturbation. This model, which is based on an off-the-shelf machine learning algorithm, allows to measure the impact of each feature on privacy, thus enabling their proper control. We observe that the features related to users' mobility and level of data perturbation are the most relevant factors affecting privacy.

- As users may be interested to exploit LBSs, they should also be aware of the impact that privacy protection has on service effectiveness. To this aim, we explore the trade-off between user's privacy and the utility of a LBS proximity marketing service. This evaluation, which is performed at varying degree of data perturbation, shows that a significant privacy improvement can be obtained at the cost of a modest deterioration of service effectiveness.

The rest of the Chapter is structured as follows. In Section 6.2, we review previous works related to information leakage on OSN with particular attention to geo-location privacy on Twitter. Section 6.4 is devoted to the methodology to measure users' geo-location privacy. Section 6.5 presents the proposed strategies to control the level of users' privacy. Experimental settings and results are presented in Section 6.6 and 6.7, respectively. In Section 6.8, we elaborate on the trade-off between privacy and utility. Finally, Section 6.9 concludes the Chapter.

6.2 Related Work

6.2.1 Privacy in OSN

There is an increasing concern on the ability of users to effectively hide personal information in OSN [118]. OSN providers include in their platforms several privacy-preserving strategies (e.g., to restrict the access to the published contents to some users only). However, it has been shown that such strategies are not fully effective in the protection of personal data in the most popular OSN [77]. In this respect, social cues (e.g., strength of social connections and similarity patterns between users)

are widely regarded as one of the main cause of privacy vulnerability in OSN. For example, it has been shown that public data on OSN can be effectively used to infer users' personal information [96, 15, 119, 28, 17, 70] and to predict future users' activity [82, 16]. Similarly to these studies, our work shows how OSN users are not in full control of their privacy [53], as sensitive information can be obtained by analyzing other users' data. In [42], authors present a generic mathematical model of attacks to violate users' privacy from publicly-available data in OSN (including location privacy). In line with these researches, we introduce a deep-learning approach specifically targeted at the violation of users' location privacy. We then propose a model that allows to assess the extent to which several factors affects the privacy of users, therefore enabling its proper tuning and control.

6.2.2 Control of Geo-Location Privacy

Although privacy-control mechanisms are offered by most social media platforms, location leakage in OSN remains an open problem [77]. For example, Polakis et. al [99] identify several vulnerabilities related to location privacy in Facebook and Foursquare and propose a set of guidelines to limit them. A theoretical framework to evaluate privacy-preserving strategies against various types of attacks in LBSs has been proposed by Shokri et al. [110]. The most adopted technique to protect location is based on the perturbation of the location. For instance, Refs [56, 21, 62] propose to reduce the spatial and temporal resolution of location traces to protect users' anonymity. In this work, we apply similar data perturbation techniques on the privacy of Twitter users. In particular, we quantitatively measure the impact on users' privacy caused by the obfuscation and by the reduction of the shared geo-tags.

The topic of privacy control has recently gained attention. In [22], the authors develop a method to measure the level of users' anonymity in LBSs and propose countermeasures to increase their privacy. Baron et al. [18] propose a framework to assess the likelihood that the public exposure of a location leads to the leakages of personal information (e.g., political view). By using this framework, users can evaluate their vulnerability to privacy attacks and understand the factors behind it. Our work shares with these previous studies the objective of giving users methods to measure and control their privacy. The main difference of our study with respect to [18, 22] is the information that we aim to protect, i.e., users' location. Moreover, given that the knowledge of location is required for the effective delivery of many services, users should be aware that privacy and quality of service are conflicting in several scenarios. This trade-off has been considered, for example, in [4] and [23], whose authors propose strategies to minimize the loss of the quality of the offered service while ensuring a minimum level of privacy in the context of video content delivery and LBSs, respectively. In this study, we perform an assessment of the trade-off between the privacy of location and the effectiveness of a LBS service delivered over an OSN platform. Differently from our work, in [4] users' privacy is managed by the video content provider and not by the users themselves. We perform a qualitative evaluation of the trade-off between utility and privacy, whereas in Ref [23], the utility is

optimized under several privacy constraints.

6.2.3 Location in Twitter

Being Twitter one of the most used OSN, the importance of both sharing and protecting location information on its platform is widely-recognized [120]. For example, various applications for emergency detection [8, 78], health monitoring [32], and events recommendation [92, 114] are based on the location information shared on Twitter. Recently, large efforts have been dedicated to the development of tools to perform location inference on Twitter. According to [120], location on Twitter can be of three main types: *home location*, *mentioned location*, and *tweet location*. The first one represents user's long-term residential address, which may be published at several levels of granularity (e.g., city or village) in the user profile. The second refers to the locations that users mention in the text of their tweets. The third is the geo-tag that users may publish as a meta-data attached to their tweets. The decision to either provide a geo-tag or not is done for each published tweet. On average, 1% of tweets is published with a geo-tag [55]. As described in [120], tweet location can be uncovered by relying on multiple sources of information: (*i*) tweet content [45, 114, 69], (*ii*) Twitter social network [103], and (*iii*) Twitter contextual information [107, 33] (i.e., meta-data related to both tweets and users' profiles).

In this work, we propose a novel deep learning architecture for unveiling users' geo-location based only on social network information. The proposed approach aims to infer the geo-tag of a generic user's tweet by only leveraging the geo-tags shared by other users on Twitter. The rationale is to investigate whether OSN users can effectively hide their location information. This intuition takes inspiration from [103], where a geo-tag published by a user is inferred considering information of her friends on Twitter (i.e., their geo-tags and the time when tweets are published). In [103], nearby locations are merged into a cluster and the location inference is framed as a classification task, where the objective is to maximize the classification accuracy. In this approach, the classification error does not carry information of the geographical distance between the target and estimated clusters. Differently from [103], in our work we measure privacy as the geographical distance between estimated and actual locations and we frame the geo-location inference as a regression problem. The objective of this regression is the minimization of the aforementioned distance, i.e., the privacy of a user. Also, to the best of our knowledge, this is the first approach that attempts to assess users' location privacy based only on the locations shared by other OSN users.

6.3 Problem Definition

In this work, we aim to infer the geo-tag of the generic user u by exploiting only the past location information of u and of the other users within Twitter. Notice that we use the terms *geo-tag* and *location* interchangeably. We consider time to be discretized into time slots of fixed duration Δt. We

refer to t_i as the time slot i. We assume that, when multiple geo-tagged tweets occur in the same time slot, we discard all but the first geo-tag.

We represent Twitter as a directed graph $G = (V, E)$, where V is the set of users and E is set of edges connecting them. On Twitter, social connections among users are based on the *followee/follower* paradigm. A generic user $v \in V$ can follow u without being necessarily followed back. For this reason, we consider u and v to be *friends* iff $(u, v) \in E$ and $(v, u) \in E$, i.e., iff both users follow each other. According to this definition, we denote the 1-hop neighborhood of u as the set of u's friends, the 2-hop neighborhood as the set of u's friends of friends, and so on. As we explain in the next Section, we utilize the concept of social proximity to select the N users in the k-hop neighborhood of u that provide their location at t_i. We refer to this set of users as *neighbors*.

Formalizing the problem, our objective is to determine the geo-location $\mathbf{l}_{t_i}^{(u)}$ of u at t_i, exploiting the known locations of u's neighbors and the location history of u, i.e., $\mathbf{l}_{t<t_i}^{(u)}$. Overall, for each user, we aim to find a function f that learns spatial and temporal dependencies of u with her neighborhood. Therefore, we define $f(\mathbf{x})$ as

$$f(\mathbf{x}) = f(\{\mathbf{l}_{t\leq t_i}^{(q)}, \mathbf{l}_{t\leq t_i}^{(r)}, \ldots, \mathbf{l}_{t\leq t_i}^{(z)}, \mathbf{l}_{t<t_i}^{(u)}\}) = \hat{\mathbf{l}}_{t_i}^{(u)} \tag{6.1}$$

where $\hat{\mathbf{l}}_{t_i}^{(u)}$ is the predicted location of u at time slot t_i, while q,r, and z indicate u' neighbors at slot i.

6.4 Geo-Location Privacy Measurement

In this Section, we initially present the formulation of the privacy measurement problem. Then, we describe the deep-learning methodology that we employ to solve it.

6.4.1 Geo-Location Privacy Definition and Measurement

As we mentioned in Section 6.1, social cues represent relevant information to violate users' privacy. We model social relationships within OSN as an undirected graph $G = (V, E)$, where V is the set of users and E is set of edges encoding friendship relations. Notice that, on Twitter, social connections are based on the *followee/follower* paradigm. Hence, we consider two users to be friends if both follow each other and we denote the 1-hop neighborhood of u as the set of u's friends, the 2-hop neighborhood as the set of u's friends of friends, etc.

In this Section, we describe the proposed approach to violate a target user's privacy from other users' data available on OSN. Privacy is generally considered a problem-dependent and subjective metric. This holds true for location privacy as well. In fact, each user may have a different perception of the intrusiveness of a precise localization. However, to perform our analysis, we require an objective measure of privacy. We define the privacy P_u of the generic user u as the average geographical distance between the locations visited by u and the ones estimated by a certain attacker. In the following formula

$$P_u = \frac{1}{N_u} \sum_{t_i} Dist(l_{t_i}^{(u)}, \hat{l}_{t_i}^{(u)}) \tag{6.2}$$

$l_{t_i}^{(u)}$ and $\hat{l}_{t_i}^{(u)}$ are the actual and estimated geo-tags of the tweet published by user u at time t_i. *Dist* is the geographical distance, i.e., the distance between two locations (expressed as pairs of latitude and longitude) measured along the surface of the earth. N_u is the total number of tweets published by u and provided with a geo-tag.

Based on this definition, we investigate whether a user's undisclosed location can be uncovered by leveraging the locations shared by other OSN users. Formally, the objective of the attacker is to estimate the geo-tag of the content (in the considered case, a tweet) published by the generic user u from the set of geo-tags that have been shared on the OSN platform by other users within a given period of time. Such period of time is discretized into time slots of fixed duration Δt and we refer to t_i to indicate the i-th time slot.

To perform this inference, we need an estimator Θ that models both the spatial and temporal dependencies between target user u and the other users within the OSN. In the following formula,

$$\hat{l}_{t_i}^{(u)} = \Theta(l_{t<t_i}^{(u)}, \mathbf{l}_{t\leq t_i}^{(\mathcal{F})}) \tag{6.3}$$

$\hat{l}_{t_i}^{(u)}$ is the estimated location of the tweet published by u at time slot t_i; $l_{t<t_i}^{(u)}$ is the set of geo-tags published by u before time slot t_i and $\mathbf{l}_{t\leq t_i}^{(\mathcal{F})}$ is the set of geo-tags published by other users up to time slot t_i. Notice that we include also the geo-tags provided by other users within time slot t_i itself because we expect it to be highly-informative of the current location of u. In the next subsection, we describe how we take into account both temporal and social proximity to design the estimator Θ.

6.4.2 Data Selection

From a theoretical standpoint, the estimator Θ could consider the whole historical information available on the OSN, i.e., all geo-tags published up to t_i. However, this amount of data grows linearly with the number of users and with the considered period of time. Hence, this approach can not scale to large instances of OSN (which can count up to several hundreds of millions of users in real scenarios). In addition, most of the geo-tags are expected to be uninformative with respect to the location of u at time t_i and can be safely discarded. In particular, tweets that have been published (*i*) far away in time and (*ii*) by users with weak social relationships with u are expected to be the least relevant ones [89]. To reduce the volume of available data, we perform a data selection process that undergoes two subsequent phases, namely the *valid slots selection* and the *valid users selection*.

In the first phase, we select the last T time slots before t_i when user u has published at least a geo-tagged tweet. Once this phase has been performed, N valid users are chosen, in each valid slot, among those who have published a geo-tagged tweet within that slot. The valid users selection phase

is carried out privileging users with high social proximity with u. Therefore, the first candidate users are those who are 1-hop-away from u. If the number of such users is less than N, the process is carried out for users that are 2-hops-away from u, etc. The process concludes when N valid users are found. We refer to this set of users as *neighbors*. It should be noticed that when it is not possible to assign u a set of N valid neighbors, e.g., in the case of disconnected network components, we assign specific missing values to $\mathbf{l}^{(\mathcal{F})}_{t \leq t_i}$, as detailed in the following subsection. We stress on the fact that other users' geo-tags are considered if provided in valid slots, i.e., when also the target user u has published at least one geo-tag. This choice is motivated by the objective of learning a consistent spatio-temporal dependency between u and her neighbors. Notice also that the N selected users differ at each time slot, as not in every time slot the same users have geo-tagged their tweets. In the next subsection, we present the deep-learning based model that we use to learn the estimator Θ.

6.4.3 Deep Learning Model for Geo-Location Privacy Measurement

The deep learning architecture proposed to learn the estimator Θ is trained separately for each user. The architecture has two main inputs, namely the information relative to the geo-tags published by the target user before time slot t_i and by her neighbors up to time slot t_i (included). The model is trained to map these inputs to the output, i.e., the geo-tag of the target user at time slot t_i. Notice that the generic geo-tag l_{t_i} is represented as a vector with 2 components, the first for the latitude and the second for the longitude. The architecture is designed to initially process its inputs separately and to perform a successive downstream elaboration of their representations. This approach is common in the machine learning community to learn an effective representation of the inputs [68]. To realize such design objective, the architecture is composed of the following four main building blocks: *i*) *Target User Transform*, *ii*) *Neighbors Aggregator*, *iii*) *Concatenator* and *iv*) *Regressor*. A representation of the architecture is depicted in Fig. 6.1. The transform and aggregator blocks perform a first processing of previous geo-tags of the target user and her neighbors, respectively; the concatenator simply juxtaposes the outputs of the previous processing and provides a single input for the regressor; finally, the downstream regressor returns the inference of the target user's location. The overall deep learning architecture is trained to minimize the Mean Squared Error (MSE) between the estimated and actual geo-tags of user u in the selected slots.

Each block of the deep learning architecture is described in detail in the next paragraphs.

6.4.3.1 Target User Transform

This block takes as input the sequences of u's geo-tags provided in the last T valid time slots, i.e., $l^{(u)}_{t_{i-T}}, ..., l^{(u)}_{t_{i-1}}$. Each geo-tag is processed separately by a feed-forward neural network, which returns an abstract representation of its input. The considered neural networks are independent, i.e., they do not share any parameter. The output of this block is a sequence of T elements containing the

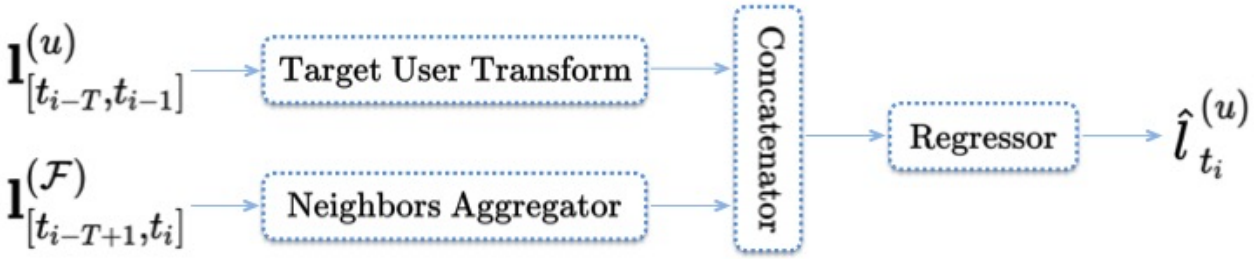


FIGURE 6.1: Overview of the deep learning architecture

representations of the locations geo-tagged by u in the selected T valid slots. The process performed by this block is depicted in Fig. 6.2a.

6.4.3.2 Neighbors Aggregator

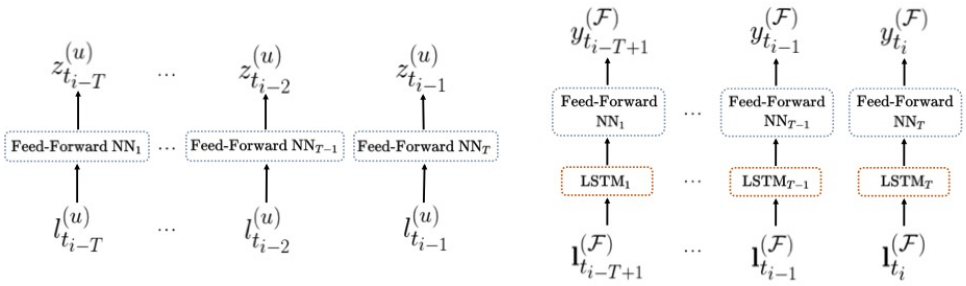
This block computes an overall representation of the information relative to the N neighbors who have been previously selected in each valid time slot, i.e., $\mathbf{l}^{(\mathcal{F})}_{t_{i-T+1}}, ..., \mathbf{l}^{(\mathcal{F})}_{t_i}$. Notice that $\mathbf{l}^{(\mathcal{F})}_{t_k}$ is the list of all the geo-tags provided by the selected users within the k-th time slot. The list corresponding to each selected slot is individually processed by a Long-Short Time Memory (LSTM) [61] and then by a feed-forward neural network. The choice of the LSTM in this phase is inspired by [59], in which it is suggested to use it as a tool to learn the representation of a node's neighbors within a graph. In our approach, topological information are considered since, as explained in Subsection 6.4.2, neighbors are chosen according to their proximity with respect to the target user. Apart from this, the LSTM guarantees higher expressive capabilities with respect to other data aggregation methods, e.g., mean aggregator [59]. The output of this block is a sequence of T elements containing the representations of the locations geo-tagged by the selected neighbors in the T valid slots. A representation of this block is depicted in Fig. 6.2b.

6.4.3.3 Concatenator

This block receives in input two sequences of T elements, which encode the representations of the geo-tags provided in the selected T slots by the target user and by the N neighbors, respectively. The concatenator block juxtaposes the elements in corresponding positions of the sequences and returns a single sequence of T elements which can be processed by the downstream regressor. A representation of this block is depicted in Fig. 6.2c.

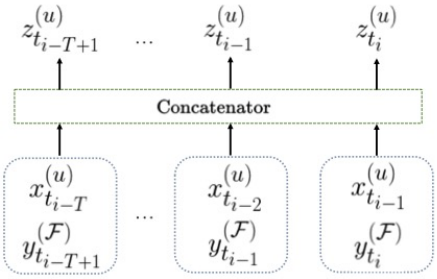
6.4.3.4 Regressor

The final block is composed of a LSTM and feed-forward neural network module. This block processes the outputs of the concatenator and returns the inferred geo-location. Notice that, even

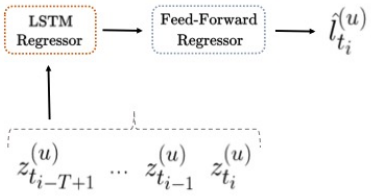


(a) Target User Transform

(b) Neighbors Aggregator



(c) Concatenator



(d) Regressor

Figure 6.2: Building blocks of the deep learning architecture

though the architecture of this block is quite similar to that of the neighbors aggregator block (see Fig. 6.2b), the design strategies behind them are significantly different. In fact, in this phase, we want to exploit the ability of the LSTM to learn recurrences within a sequence of temporal data, which are useful to perform the inference task. A representation of this block is depicted in Fig. 6.2d.

The proposed deep learning architecture returns an estimate location $\hat{l}_{t_i}^{(u)}$ for each geo-tagged tweet shared by user u. Finally, the privacy of u is measured according to Eq. (6.2), i.e., by computing the geographical distance between actual and estimated locations.

6.5 Control of Privacy Level

After having shown how to measure user's geo-location privacy in Section 6.4, in this Section we describe two strategies that users can implement to enhance their privacy, i.e., to reduce the ability of an attacker to correctly infer their location. Then, as the measured privacy is likely to be affected by other factors beyond data perturbation, we also propose a privacy model that captures the impact on privacy of several users' behavioural characteristics (e.g., data perturbation level and users' mobility) and that can therefore be employed as a more comprehensive privacy control tool.

6.5.1 Strategies to Tune the Level of Privacy

Every time a user publishes a tweet, she can decide to either provide it with a geo-tag or not. In this respect, each user is characterized by a particular behaviour, which can be defined as the percentage of tweets that she normally geo-tags. As location privacy can be violated by relying on the information released by other OSN users, to prevent this violation and preserve the secrecy of their location, users may purposely alter the geo-tags of a portion of the tweets they normally publish. We refer to this strategy as *data perturbation* and we introduce a variable p, the data perturbation probability, i.e., the probability that a user deviates from her normal behaviour. For instance, a user who publishes, on average, 10 geo-tags per month, can set $p = 0.3$ and reduce the number of normally geo-tagged tweets to 7. The remaining 3 tweets are perturbated. We propose to use two data perturbation strategies, namely *data obfuscation* and *data removal* strategy and we describe them in the following.

6.5.1.1 Data Obfuscation

According to this strategy, a geo-tag is shared, with probability p, by randomly selecting a location within the boundaries of the city where the user is tweeting (New York City, in our dataset).

6.5.1.2 Data Reduction

Following the data reduction strategy, with probability p, a user does not geo-tag a tweet that would have been provided with a location if no perturbation were applied.

The rationale is that, by increasing p, i.e., the level of data perturbation, a user can improve her privacy. However, we shall still provide a quantitative answer to an important pending question: how much privacy a user should expect to gain by increasing p?

A user who is interested to have a quantitative assessment of her privacy can follow the approach described in Section 6.4.3 to infer her visited locations (as an attacker would do) and, from this, evaluate her own level of privacy. This approach is effective to estimate users' expected level of privacy, but has several drawbacks. First of all, to assess the impact of the data perturbation level on the resulting privacy, the user has to train and test the deep learning model using data perturbated

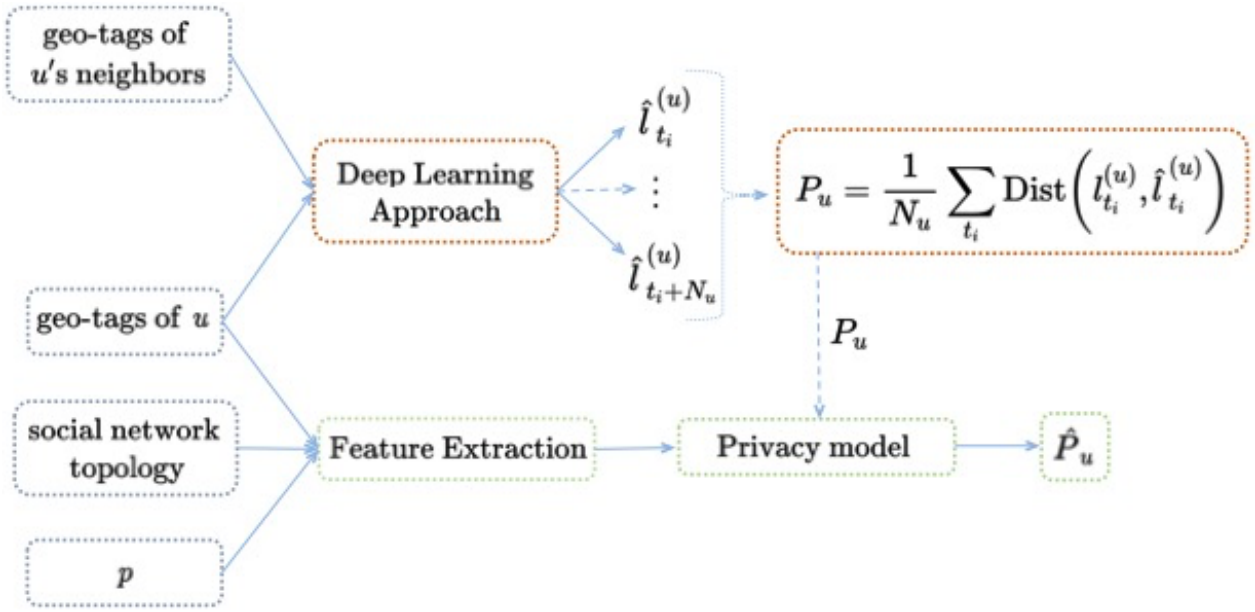


Figure 6.3: Block diagram of the approach followed to learn the Privacy Model. Notice that P_u is the target value that the privacy model aims to estimate.

for several values of p. This process, besides being highly-time consuming, does not allow users to understand the impact that several factors (e.g., related to her mobility behaviour) beyond p have on her privacy. In fact, this model takes as input only raw data (i.e., geo-tags) that do not explicitly capture the characteristics of users' behaviour. Moreover, there is no possibility to interpret the impact of each feature on privacy, since the output of the model is itself a location (and not a value of privacy). Lastly, deep learning models are widely considered as black boxes whose outcomes are difficult to interpret in relation to the input features [57, 79]. Hence, the proposed deep learning approach is only capable to measure privacy, but it does not give users a proper understanding on how to control it. In the next subsection, we describe a *privacy model* that quantifies the impact of various factors on privacy and allows to directly measure its level (i.e., without performing an intermediate location inference).

6.5.2 Privacy Model

In this subsection we present a privacy model that provides users with an estimate of their privacy level given a set of features that explicitly capture their characteristics and behaviour. This model is based on an off-the-shelf machine learning algorithm trained in a supervised manner to map diverse users' features to their value of privacy. Machine learning algorithms are commonly trained using ground truth values. In our case, however, the target privacy cannot be regarded as a ground truth value in the traditional sense. In fact, privacy is not an attribute of the users, but rather a value

derived from the estimation of their location, which in turns depends on many factors (e.g., ability of the attacker, amount of available data, etc...). In this study, we consider as target value the measurement of privacy obtained with the deep learning approach explained in Section 6.4.3, as represented in Fig. 6.3. Notice that, in principle, any estimation of privacy could be used as target value.

We train several off-the-shelf machine learning algorithms (e.g., random forest and decision tree) to obtain an estimator of users' privacy (results in Section 6.7.3). These algorithms are trained to minimize the Mean Absolute Error (MAE) with respect to the target privacy value. The MAE is defined as:

$$MAE = \frac{1}{|V|} \sum_{u \in V} |P_u - \hat{P}_u|, \tag{6.4}$$

where P_u and $\hat{P}_u$ are the target privacy value of user u (computed according to Eq. (6.2)) and the privacy value estimated by the off-the-shelf model, respectively. $|V|$ is the total number of users present in our dataset.

To obtain the privacy model, such off-the-shelf algorithm is fed with a set of features, which are listed in Table 6.1. These features cover a broad range of parameters that users can tune to control their privacy, and fall into three main categories, namely *mobility-related*, *topology-related*, and *data-perturbation-related*. Mobility-related features are considered to statistically describe the mobility (e.g., in terms of variability of the visited locations [94]) of the users. Topology-related features aim to provide a characterization of users' position within the social network (e.g., in terms of network centrality measures). The data-perturbation-related feature is p, i.e., the level of data perturbation the user is willing to adopt for tuning her privacy.

Given these features for a certain user, the model outputs an estimate of her location privacy. If, for example, a user is not satisfied with her privacy level, she can evaluate how the secrecy of her information enhances by varying her sharing activity (e.g., by geo-tagging less frequently), changing her social clique (e.g., by diminishing the number of friends on the OSN), or perturbing their geo-tag (i.e., by increasing the data perturbation probability p).

Table 6.1: Features of the user's geo-location privacy model

Category	Feature	Description
Mobility-related	tweet frequency	Frequency of geo-tagged tweets per day
	avg. distance	Average distance of geo-tags
	var latitude (longitude)	Variance of the geo-tags latitude (longitude)
	corr. latitude (longitude)	Autocorrelation of the geo-tags latitude (longitude)
	median latitude (longitude)	Median of the geo-tags latitude (longitude)
	kurtosis latitude (longitude)	Kurtosis of the geo-tags latitude (longitude)
	skew latitude (longitude)	Skewness of the geo-tags latitude (longitude)
Topology-related	no. of friends	Number of friends
	PR	PageRank
	deg-centrality	Degree centrality measure
	eig-centrality	Eigenvector centrality measure
	cl-centrality	Closeness centrality measure
	bw-centrality	Betweeness centrality measure
Data perturbation-related	p	Data perturbation probability

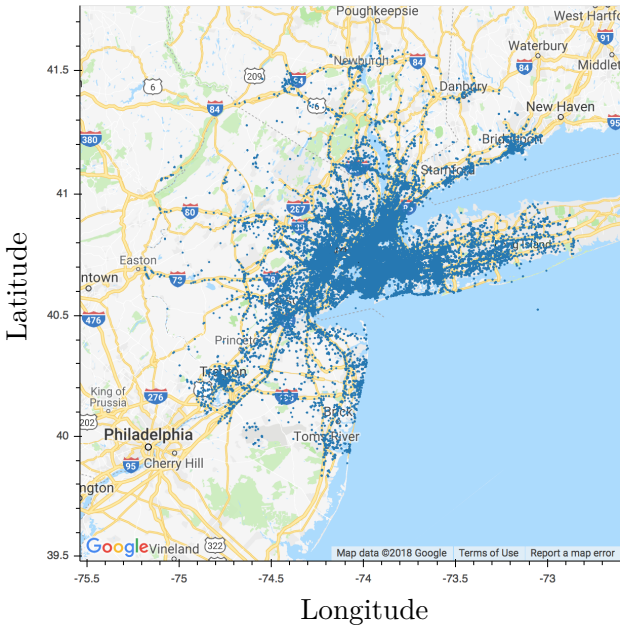


FIGURE 6.4: Geographical Distribution of the Geo-tagged tweets in New York City

Table 6.2: Statistics of the Twitter dataset

	New York City Dataset
Unique users	6082
Friendship relationships	31874
Average degree	10.22
Diameter	19
Clustering coefficient	0.15
Density	0.001

We envision the utilization of this model in a client-server scheme, where the client is a user who requires an estimate of her privacy, while the server is a third-party entity that offers a service of privacy awareness. Notice that this approach is also privacy-preserving, since it requires users to disclose to the server only the aforementioned features, which, as detailed in Table 6.1, represent aggregate information (e.g., variance of their visited locations) and do not need the sharing of the visited locations.

6.6 Experimental Setup

6.6.1 Data

We perform our evaluations on the Twitter dataset presented in [103], which includes the information about social connections among users (i.e., pair of followees, followers) and $2,173,681$ tweets collected within 100 kilometers from New York city center for 31 days. Figure 6.4 shows the spatial distribution of the tweets over all the collection period. In Table 6.4, we summarize the statistics about the data and network properties related to the social graph. The *average degree* is the average number of friends over all the users in the social network, while the *diameter* is the longest of the shortest paths in the social network. The *clustering coefficient* is the average of the clustering coefficients over all the users, i.e., the ratio between the number of links connecting user's friends to each other and the total number of possible connections. Finally, the *density* is the ratio between the number of edges connecting the users and the total number of edges in the network.

6.6.2 Simulation Settings

We evaluate the privacy of a user (defined as the geographical distance between her actual and estimated location) at a given time slot t_i, given the information of the geo-tags published on the past T slots. After preliminary evaluations, we found that $T = 3$ was the best compromise between accuracy and training time of the deep learning algorithm. Each slot represents a period of 3 hours, which is the average time between two consecutive tweets in the dataset. Each location published during the slots is expressed as a pair of latitude and longitude. Both latitude and longitude have been normalized according to the standardization technique, which is widely employed to perform feature scaling in machine learning.

The deep learning architecture includes a *target user transform* (i.e., a feed forward neural network composed of a single layer with 5 neurons), a *neighbors aggregator* (i.e., a LSTM layer with 5 neurons followed by a feed forward neural network layer with 2 neurons) and a regressor (i.e., a LSTM layer with 5 neurons and a feed forward neural network layer with 2 neurons). Among the configurations of hyper-parameters that we have considered, the aforementioned one proved to achieve the best balance between inference effectiveness and training time efficiency. The deep learning model is trained to estimate the location of the target users at time slot t_i from the previous T geo-tags provided by the target user u (up to time slot t_{i-1}) and those of the N neighbors (up to time slot t_i) chosen in the valid users selection phase. The number of neighbors N is set to 10, that is the network average degree (see Table 6.4). The first 75% of the time slots are used for training purposes, while the remaining 25% are used to test the learned model. The 75% of training slots are further divided into pure training slots (60%) and validation slots (40%).

Table 6.3: Comparison between the deep-learning and the baseline approaches

Approach	Average Error [km]
Deep Learning	2.3
Median-Based	7.3
Mean-Based	7.7
Cluster-Based	9.2

6.7 Results

6.7.1 Privacy Measurement

In this Section, we present an overview of the results related to the privacy measurement obtained by using the proposed deep-learning approach described in Section 6.4.3. To motivate the use of a machine learning strategy, we firstly show the average location inference error obtained using several alternative approaches. In particular, we propose three baselines that perform the location inference of a target user based on the same inputs of our deep-learning algorithm, i.e., her most recent available location and the geo-tags of other users. Specifically, the three baselines compute a value $\hat{\mathcal{F}}_{loc}$ from the geo-tags of $N = 10$ target users' neighbors as follows:

- *mean-based*: $\hat{\mathcal{F}}_{loc}$ is obtained by computing the average latitudes and longitudes of the N neighbors.
- *median-based*: $\hat{\mathcal{F}}_{loc}$ is obtained by computing the median value of latitudes and longitudes of the N neighbors.
- *cluster-based*: the geographical area considered in our dataset is divided into a set of non-overlapping squares with sides equal to $\sim$ 155m. $\hat{\mathcal{F}}_{loc}$ is the centroid of the square mostly visited by the neighbors.

The location of the target user is then estimated by computing the average between her most recent available location and $\hat{\mathcal{F}}_{loc}$. In Tab. 6.3, we show the average error of the location inference obtained using the deep-learning approach and the three baselines. As expected, the deep-learning algorithm significantly outperforms all the considered alternatives, which confirms its ability to capture complex relationships among the input data.

To further elaborate on the results obtained with the deep-learning strategy, we show the distribution of the location privacy in Fig. 6.5. It can be noticed that 60% of the users have a level of privacy below 1km and almost 80% of the users achieve a privacy measure below 3km. Overall, the average privacy measure is of 2.3km and the median privacy value is of 0.4km. Such results corroborate our intuition that the secrecy of the location information can be violated by leveraging publicly available information shared by other users.

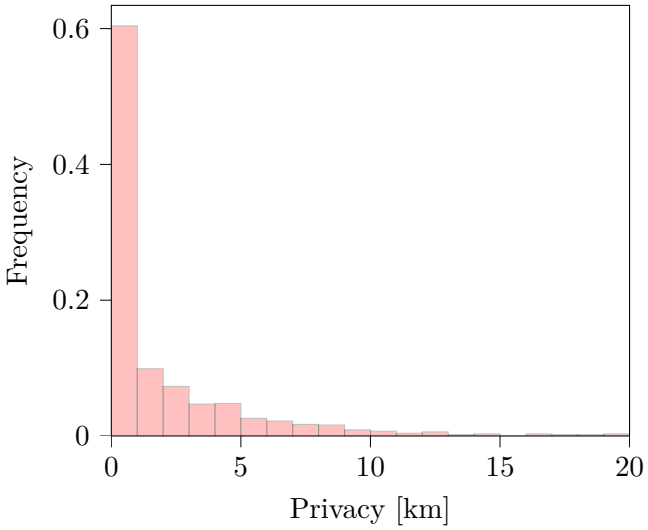


Figure 6.5: Distribution of the location privacy measurement

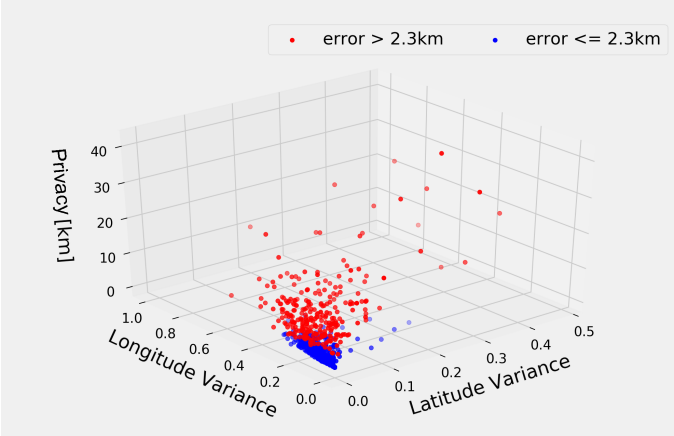


Figure 6.6: Geo-localization error vs Variance of Mobility. Users whose localization error is lower (resp. higher) than the average (2.3km) are represented as blue (resp., red) points.

To better understand this result, in Figure 6.6, we depict the privacy measure as a function of variances of latitude and longitude visited by each user. Users whose privacy is lower (resp. higher) than the average (2.3km) are colored in blue (resp., red). We expected that the variance of the locations visited by the user could be a strong indicator of user's privacy leakage. However, a subset of users with limited mobility achieves a privacy level above the average, suggesting that mobility variance is not the only factor determining the level of privacy risks. We further analyze other factors impacting users' privacy in Section 6.7.3.

6.7.2 Data perturbation strategy

The results related to the privacy assessment indicate a considerable peril for users' privacy in OSNs. We now evaluate the effect of two possible countermeasures to increase the location error of attacker's estimates. Thereby, we examine the data perturbation strategies introduced in Section 6.5.1. In Fig. 6.7, we depict the percentile of the geo-location error with varying data perturbation probability p related to the data obfuscation strategy. Notice that the geo-location error corresponds to the definition of privacy provided in Section 6.4.1. We firstly observe that the percentage of perturbed geo-tagged tweets significantly affects the ability of the attacker to correctly infer the geo-tags. In fact, as expected, the localization error increases with increasing p, i.e., users' privacy increases as the level of perturbation increases.

On the other hand, as shown in Fig. 6.8, the data reduction strategy does not improve users' privacy as much as the data obfuscation strategy. The gap between the percentiles with varying p is very small, i.e., there is no significant privacy improvement with respect to the unperturbed scenario ($p = 0$). This result suggests that even a small percentage of uncorrupted information is sufficient to effectively estimate users' location. This is further confirmed in Fig. 6.9, which compares the average localization error of the perturbation strategies with varying p. Interestingly, in the data obfuscation strategy, users' privacy grows linearly with p, whereas the localization error is almost constant around 3.5km using the data reduction strategy. We observe that, using the data obfuscation strategy, privacy increases proportionally with the data perturbation level, which can in turn be tuned according to the desired level of privacy.

This result suggests that data obfuscation is preferable with respect to data reduction. However, several issues are still worth exploring to further validate this outcome. For instance, the robustness of data obfuscation to eventual counter-reactions of the attacker should be carefully evaluated. In fact, an attacker might identify the altered geo-tags and discard them to perform the location inference. Intuitively, the ability to discriminate between actual and altered geo-tags grows with increasing the area in which users provide their obfuscated locations (as it becomes more evident when a user publishes geo-tags far away from her common visited locations). However, a thorough evaluation of the data obfuscation strategy is out of the scope of this work and it is left as a future study.

6.7.3 Validation of the privacy model

In this subsection, we show the results on the privacy model described in Section 6.5.2. Specifically, we evaluate the ability of this model to estimate users' privacy and we examine the impact that users' behavior have on their level of privacy. To this end, we train and compare several off-the-shelf machine learning algorithms. In particular, we train and test all the algorithms by following a 10-fold cross validation approach.

Table 6.4 shows the MAE for the different considered approaches. For further evaluations, we

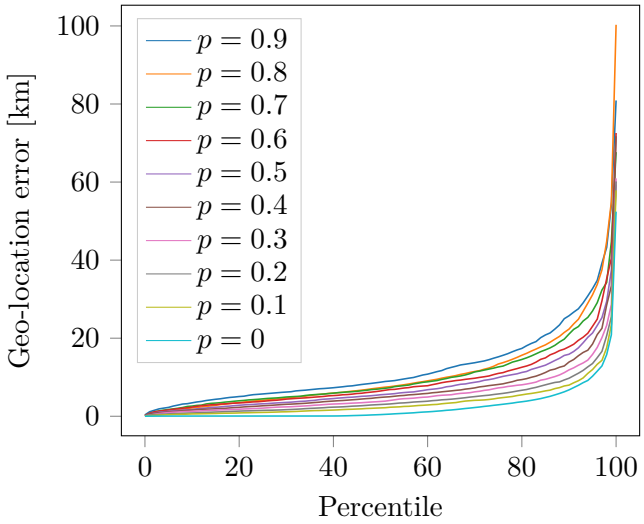


Figure 6.7: Percentile of the geo-localization error using the data obfuscation strategy with varying p

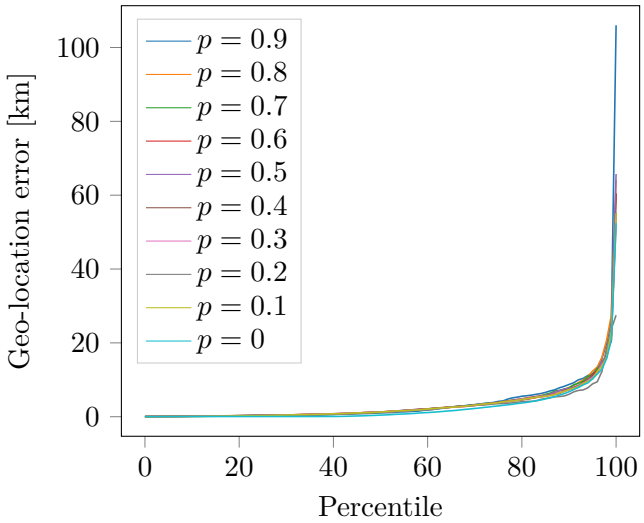


Figure 6.8: Percentile of the geo-localization error using the data reduction strategy with varying p

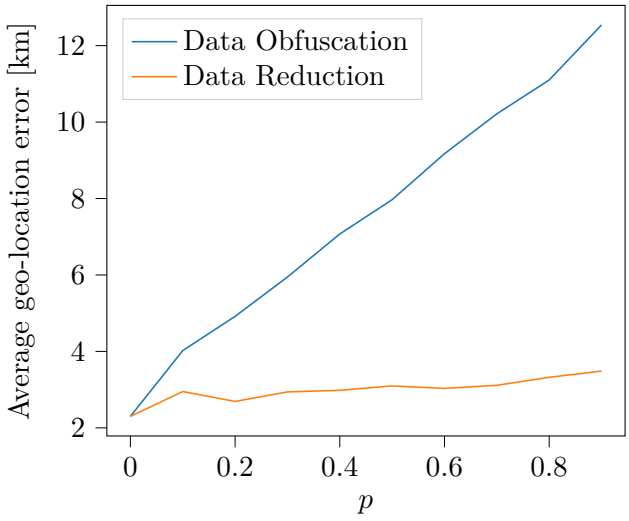


FIGURE 6.9: Comparison of the data perturbation strategies considering the average geo-localization error with varying p

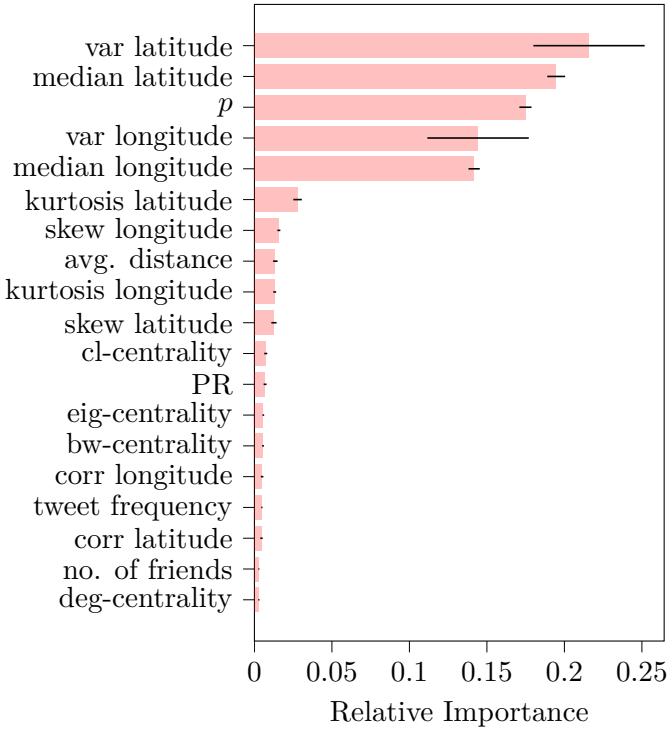
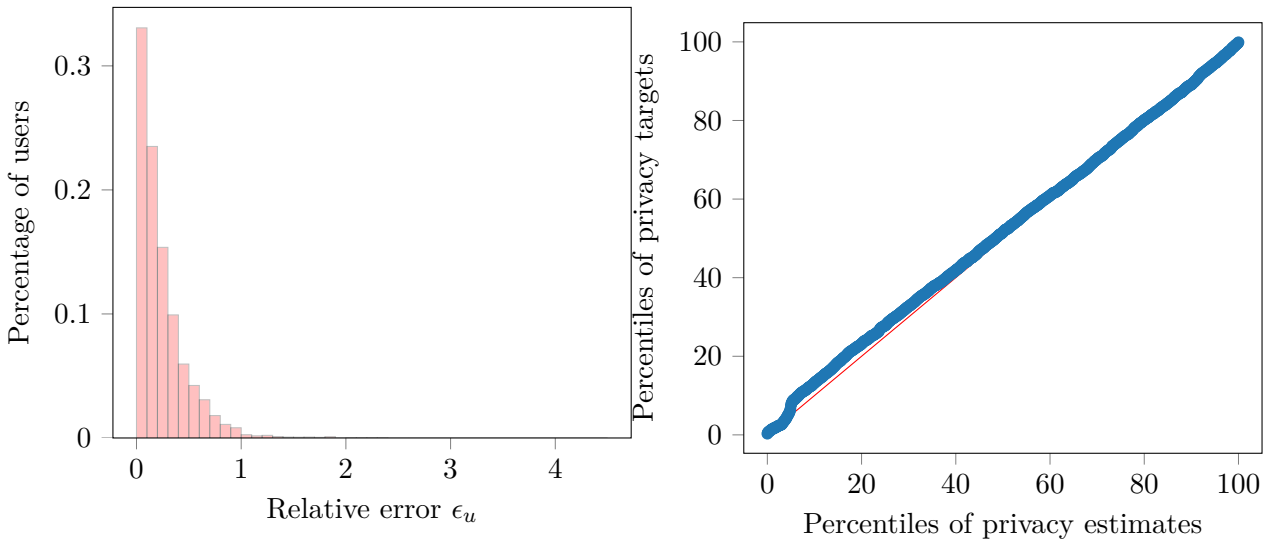


FIGURE 6.10: Features importance of the model based on random forest

consider the model that yields the minimum MAE, i.e., the model based on a random forest. We present the feature importance in Fig. 6.10. First of all, we observe that mobility-related features,



(a) Distribution of the relative error ϵ_u between estimates and target values of privacy

(b) Q-Q Plot

Figure 6.11: Graphical validation of the privacy model

Table 6.4: Performance of the privacy model for several machine learning algorithms

Algorithm	Error [km]
Random Forest	1.4
Decision Tree	2.1
Gradient Boosting	3.0
k-nearest neighbors	3.7
Ridge Regression	3.8
Support Vector Machine	4.1

e.g., variance and median value of the visited locations, play the most relevant role. This can be explained considering that mobility patterns strongly affect the predictability of the visited locations. Perhaps surprisingly, the median of the visited locations highly influences the outcome of the model. This may suggest that some locations are strong indicators of the privacy of a user [18]. It is also noticeable that the variables (e.g., variance and median) related to the visited latitudes have a larger impact on the outcome with respect to the variables related to the visited longitudes. Arguably, this is due to the nature of users mobility within the specific urban area under analysis. Another very relevant feature is the level of data perturbation, i.e., p. This confirms the discussion done in Section 6.5.1, where the importance of tuning p to increase privacy has been highlighted. Finally, we notice that topology-related features have the least significant impact on the estimation of privacy, suggesting that the privacy of a user is quite unrelated with her position in the social network.

Until now, we have considered the MAE between estimates of privacy (i.e., $\hat{P}_u$) and corresponding reference privacy values (i.e., P_u) as the only metric to evaluate the goodness of the privacy model. However, the MAE does not consider the impact of each estimate's error on its actual reference privacy value. For example, an absolute error of 1km is much more significant if $P_u = 3$km with respect to the case where $P_u = 50$km. To provide a more complete evaluation of the actual ability of the model to correctly estimate privacy, we propose two model validation strategies. The first one is the analysis of the *distribution of the relative errors* between estimates and target privacy values. The second one is the *Q-Q plot*, which graphically shows the likelihood that two populations have been generated by the same model. The two validation strategies are explained in details in the following.

6.7.3.1 Distribution of relative errors

We now present the distribution of relative errors between the target privacy values and the privacy estimates computed by the random-forest model. The relative error of the privacy of user u, i.e., ϵ_u is defined as:

$$\epsilon_u = \frac{|P_u - \hat{P}_u|}{P_u}. \tag{6.5}$$

We depict the distribution of the relative error in Fig. 6.11a. Notice that the relative error provides much more valuable information with respect to the MAE shown in Table 6.4, because it considers the error in relation to the reference privacy value (i.e., P_u). From Fig. 6.11a it is possible to notice that most of the users present a low relative error. More specifically, for around 33% of the users, the random forest estimates the privacy with a relative error below 10%. This percentage grows to 50% if a relative error below 20% is considered.

6.7.3.2 Q-Q plot

The *Q-Q plot* allows to graphically assess the likelihood that two populations have been generated by the same model. In our case, the elements of the two populations correspond to the privacy values estimated by the random forest model and the target values, respectively. Each element of the population is represented in the Q-Q plot as a point whose coordinates are the corresponding percentiles of the two populations. If the data were generated from the same model, each point would lie on a line of slope equal to 1 passing through the origin. In Fig. 6.11b, it is possible to observe that most of the points lie in the proximity of this line. The obtained results suggest the use of the privacy model as a reliable tool to estimate privacy of users from their characteristics.

6.8 Trade-off between Utility and Privacy

In this Section we consider the case study of a proximity marketing service delivered on an OSN platform to quantitatively evaluate the trade-off between utility and privacy. This LBS geo-localizes its users and advertises products that are sold close to them. Following [88], we define *utility* the probability that a user clicks on the advertised products and we refer to this to as $\mathcal{U}$. According to [88, 65, 83], this utility depends on many factors that interplay with each other. Among them, the geographical distance between target user and the retailer is one of the most important, as utility decreases with increasing this distance. Our analysis is based on the numerical evaluation performed in [88], which provides an estimate of the probability of a click as a function of the distance between user and advertised object. To give an example, according to [88] the probability that a user clicks on the advertisement of a product sold 0km away from her is 0.011. This probability drops to 0.008 when the distance is increased to 10km.

Exploiting our definition of privacy (i.e., the geographical distance between the actual and estimated users' location) and the relation between click probability and distance [88], we perform an evaluation of the degradation of utility with increasing privacy. Our overall objective is to assess the extent to which the effectiveness of LBSs and the protection of location privacy conflict with each other. More specifically, in our evaluation, we assume that the LBS geo-localizes its users by employing the approach presented in Section 6.4.3. Since users are likely to be interested in products that are sold in their proximity [88, 75], their utility highly depends on the ability of the LBS to perform an accurate localization.

However, users might be interested to experience a high utility while not sacrificing their privacy. Hence, it is crucial to quantitatively assess the deterioration of their utility caused by the application of data perturbation strategies. In the following, we explore this trade-off between privacy and utility as a function of the data perturbation level p, assuming that users apply a perturbation to their geo-tags using the data obfuscation approach described in Section 6.5.1.

We collect the actual and estimated locations of all the tweets that have been used to assess users' privacy by means of the deep learning architecture (i.e., test data). For the tweet published by user u at time slot t_i, we then perform 10 experiments in which we simulate the advertisement of a product located within an area of radius η centered in the estimated location $\hat{l}_{t_i}^{(u)}$. A representation of this process is depicted in Fig. 6.12, where p_1 and p_2 are products advertised at a distance of d_{p_1} and d_{p_2}, respectively, from the target user's actual position. The position of the product is assumed to follow a uniform distribution within the considered area. In each simulation, we compute the distance between the actual position of the user (i.e., $l_{t_i}^{(u)}$) and the advertised product. Intuitively, this distance mostly depends on the geo-location error, i.e., user's privacy. Based on this distance, we compute the value of the corresponding utility experienced by u, according to the analysis done in [88].

Our first objective is to evaluate the trade-off between privacy and utility as a function of the

data perturbation level p. Experiments are performed by simulating the advertisements of products within a radius of $\eta \in [1, 5, 10]$km from the estimated user's location. We present this result in Fig. 6.13. We observe a linear increase (resp., decrease) of privacy (resp., average utility) with increasing p. Then, we also notice that the utility is lower for higher values of η, because the average distance between user and product increases when the product is advertised on larger areas. Moreover, we notice that the gap among the utilities (for several values of η) decreases for high values of data perturbation. This can be explained considering that, when p is high (e.g., close to 1), the LBS severely mis-locates its users and the products are likely to be advertised in areas far away from them.

The second objective is to measure the negative impact that guaranteeing privacy has on the utility. The considered metric is the percentage loss with respect to the utility that user u would experience if she was perfectly geo-localized. To this end, we also simulate the advertisement of a product randomly placed within a circular area of radius $\eta = 1$km centered in the location where user u is actually located, i.e., $l_{t_i}^{(u)}$. The percentage loss relative to the tweet published by u at time slot t_i is defined as:

$$\mathcal{L}_{t_i}^{(u)} = 100 \cdot \frac{\mathcal{U}(l_{t_i}^{(u)})}{\mathcal{U}(l_{t_i}^{(u)}) - \mathcal{U}(\hat{l}_{t_i}^{(u)})}, \tag{6.6}$$

where $\mathcal{U}(\hat{l}_{t_i}^{(u)})$ and $\mathcal{U}(l_{t_i}^{(u)})$ are the utilities experienced by user u if she is localized with an error (privacy-preserving scenario) and if she is perfectly localized (privacy-intrusive scenario), respectively. In Fig. 6.14, we show the percentile of $\mathcal{L}$ for several values of data perturbation level p. As a benchmark, we consider the case when users do not perform any perturbation to their geo-tags ($p = 0$). In this situation, the loss with respect to the case of perfect localization is $< 2\%$ for 50% of the advertisements. When $p = 0.1$ and $p = 0.9$, 50% of the advertisements' products reach a loss $< 6\%$ and $< 22\%$. We consider this loss acceptable compared to the gain on privacy induced by data perturbation. For instance, when no perturbation is applied ($p = 0$), 50% of the users can be localized with an error < 0.4km, while for $p = 0.9$, 50% of the users can be localized with an error < 8.6km. Further evaluations are needed to extend these results to other more general use cases. However, the obtained results suggest a coexistence of privacy-preserving strategies and LBSs, as the effectiveness of the latter is robust to inaccurate users' geo-localization. Hence, users can preserve their location privacy while not significantly affecting the delivery of LBSs.

6.9 Conclusions

In this Chapter, we focused on the problem of geo-location privacy on OSNs, considering Twitter as a study case. We address this problem from two different angles: on one side, we develop methods to assess the ability of an attacker to correctly infer users' locations; on the other side, we propose effective strategies that users can adopt to measure and control their privacy.

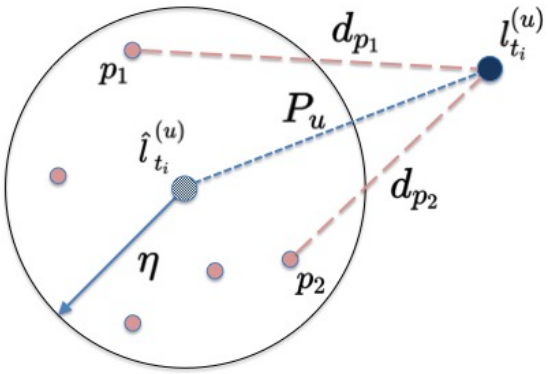


Figure 6.12: Proximity marketing in a privacy-preserving scenario: user u, which is in the location $l_{t_i}^{(u)}$ at time slot t_i, receives advertisement of a product p_i located within an area of radius η centered in the estimated location $\hat{l}_{t_i}^{(u)}$

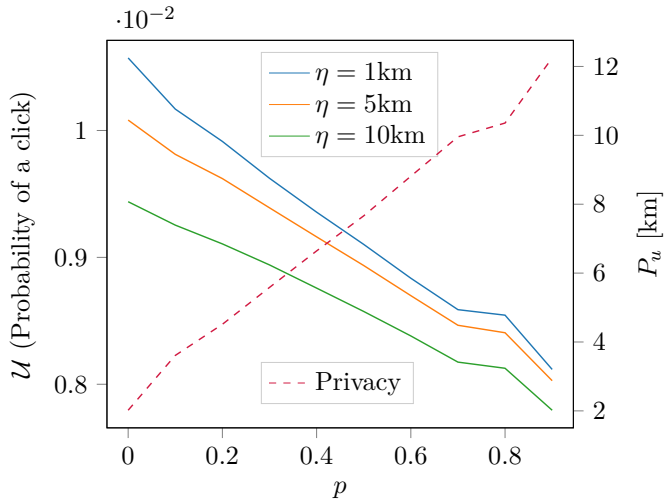


Figure 6.13: Trade-off between Privacy vs. Utility

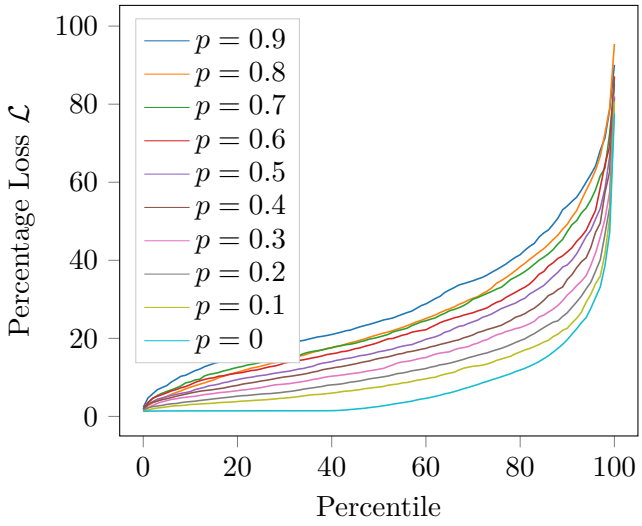


FIGURE 6.14: Percentile of the Utility Function loss with varying p

To pursue the first objective, we propose a deep learning model that can accurately infer users' location by only considering publicly available geo-tags. From this estimate, we measured location privacy as the geographical distance between the inferred and the actual position. Our experiments confirm the concerns about privacy perils in OSN. In fact, we showed that 60% of the users have a level of location privacy below 1km and almost 80% of the users achieve a privacy measure below 3km.

To achieve the second objective, we propose data perturbation techniques that users can use to decrease the knowledge obtainable by analyzing their published geo-tags. We measure the effectiveness of such strategies considering their ability to increase users' privacy. In particular, the obfuscation of the actual location proved to be the most effective strategy. To further increase users' control, we model privacy considering features that capture users' behaviour (e.g., characteristics related to the mobility of the user and to the enforced level of data perturbation). This model, based on the random forest algorithm, provides an accurate estimate of privacy and enables a principled understanding of the features that mainly affect it. Features related to the mobility of the user and to the enforced level of data perturbation resulted to be the most relevant factor behind the infringement of users' location secrecy.

Finally, we consider as a study case the effectiveness of a LBS, i.e., proximity marketing. We notice a trade-off between privacy and utility of the LBS when users enforce data perturbation strategies. We measure this trade-off and we conclude that users can achieve a significant level of privacy at the cost of an acceptable deterioration of utility. We believe that the trade-off between utility and privacy is of paramount importance also in other contexts (e.g., video content delivery) that we aim to explore in future works. Also, as the proposed methodology to measure and control privacy is based only on the geo-tag information, it is agnostic to the considered OSN platform.

Hence, we plan to extend this work by validating the proposed methodology on other OSNs, as well as on diverse types of sensitive data.

Privacy-Preserving Reinforcement Learning for Multi-domain Virtual Network Embedding 7

In this Chapter, we consider the problem of performing Virtual Network Embedding (VNE) over a multi-ISP infrastructure in a privacy-preserving manner. More specifically, we consider a problem in which a customer is willing to minimize the deployment cost of a Virtual Graph (employed to offer a service to its users) performed over the infrastructures of several independent ISPs. In this context, ISPs are concerned of exposing to such customer sensitive infrastructural details that, however, are needed to perform an effective deployment. Following a common privacy-preserving approach, the embedding may be performed by the customer based on the abstract view of the multi-domain infrastructure that ISPs accept to expose, i.e., Limited Information Disclosure (LID). With this approach, embedding is sub-optimal (e.g., embedding cost is not minimized) in comparison with the case where all information is available, i.e., Full Information Disclosure (FID). In this Chapter, we present a Reinforcement-Learning-based algorithm able to process data that customer and ISPs cipher under the Shamir Sharing Scheme (SSS) scheme. We perform extensive simulations to evaluate the effectiveness of our RL algorithm considering the total embedding cost, compared with that obtained by applying two existing LID and FID existing heuristics.

7.1 Motivation

The decoupling of the software implementation of a service from its underlying hardware, known as function virtualization [49], brings several advantages, such as increased service flexibility and scalability, as well as reduced Capex and Opex expenses. A virtualized service can be represented as a Virtual Graph (VG), i.e., a set of Virtual Nodes (VNs) and relative Virtual Paths (VPs), that a third-party entity (e.g., a *customer*) supplies to its users exploiting the physical infrastructure of an

Internet Service Provider (ISP).

The problem of embedding this graph into the physical network towards some optimization objective (e.g., minimization of the deployment cost) is referred to as Virtual Network Embedding (VNE) and has received considerable attention in the literature [49]. In particular, as VNE is proved to be NP-hard [26], many heuristics have been proposed to efficiently solve it [11]. Although VNE is a well-studied problem when a single network operator is considered, to the extent of our knowledge, the scenario where the underlying infrastructure is composed of several independent ISPs' networks has not been explored as much. In this case, the customer may benefit from an extended covered geographical area, an increased heterogeneity of the available infrastructure and, eventually, also from reduced embedding costs.

However, the multi-domain scenario introduces new challenges that increase the difficulty to effectively solve the VNE. In fact, ISPs may be concerned about the exposition of business-critical and privacy-sensitive details of their networks that are required the execute embedding algorithms. A viable approach proposed in the literature [41] to guarantee ISPs' privacy requirements while not preventing embedding consists in limiting the information available to the entity performing the optimization. Following this approach, referred to as Limited Information Disclosure (LID), the customer optimizes the assignment of portions of the VG to each involved ISP based on an abstract view of the multi-domain infrastructure (e.g., only the peering links and the cost of traversing them are visible to the customer) and, based on this decision, each ISP embeds the assigned sub-graph on its infrastructure. The main drawback of the LID approach is a sub-optimal embedding with respect to the Full Information Disclosure (FID) counterpart, in which the customer performs the optimization based on a complete view of the substrate multi-domain infrastructure.

In this study, we initially propose a multi-agent RL algorithm that customer and ISPs can execute in a distributed manner to solve the VNE problem and that ensures several privacy guarantees. More specifically, we subdivide the VNE problem into several main sub-tasks and we define a RL environment for each of them. Customer and ISPs perform operations on their environments and, based on that, receive rewards through which they learn how to efficiently solve the associated sub-task. To make the operations done towards a common optimization objective, rewards are specifically designed and properly exchanged among the participants. We compare our RL-based algorithm with the LID and FID heuristics proposed in [41] considering the overall embedding cost. Our approach generally achieves a cost that is slightly lower then the cost obtained with FID, and significantly lower than the cost obtained with LID. However, whilst the rewards are designed to provide only aggregated information about participants' data (which makes the approach somehow privacy-preserving), their exchange may still leak sensitive information.

Aiming to achieve total privacy, we then propose a privacy-preserving version of the RL algorithm that is based on the Shamir Secret Sharing (SSS) scheme. By using this approach, customer and ISPs only learn information relative to the final embedding and no sensitive information is leaked during the optimization process (e.g., the computational demand of a VN is only disclosed to the

ISP selected to host it). The main drawback is the high volume of data that participants exchange with each other to execute the privacy-preserving algorithm. We address this issue by reducing the number of expensive operations (in terms of introduced data overhead), and we evaluate the corresponding increase of embedding cost for several levels of this relaxation. Results show that, when this relaxation is low, the privacy-preserving RL generally outperforms the LID approach while achieving a reduction of the overhead of at least two orders of magnitudes.

7.2 Related Work

The topic of VNE has been extensively considered in the literature [49, 87]. In particular, several heuristics have been proposed to solve it efficiently, as reviewed in [26]. To our knowledge, smaller attention has been devoted to the VNE problem in the multi-domain scenario, in which the optimization is hindered as, to protect their privacy, ISPs do not expose sensitive information needed for the optimization. Two main approaches have been proposed in the literature to solve the multi-domain VNE problem, i.e., the distributed and centralized ones.

Examples of the former category are the works presented in [105],[117]. In [105] ISPs address privacy issues by exposing information only to other network operators they have a mutual agreement with. The main drawbacks of a distributed approach is that the optimization is not performed based on a global view of the overall network. On the other hand, existing centralized approaches generally divide the VNE problem in two sub-tasks: in the first, a VG is partitioned over the participants ISPs; in the second, each ISP performs the VNE of the received portion of graph. The first sub-task is executed by a centralized entity, e.g., a customer or a broker acting on behalf of it [63, 40, 58]. In these approaches, privacy issues are addressed as the first sub-task is executed based on the limited information about network infrastructures that the ISPs provide to the centralized entity. For example, in [40, 41] only the peering links and the cost of embedding a given VN on a physical peering node are exposed. The main drawback of this approach is that this reduction of available information leads to a sub-optimal solution of the VNE. In our work, we propose a RL-based method that achieves better embedding performance with respect to such limited-information approaches, while guaranteeing total privacy.

RL has been widely used as a tool to perform optimization in telecom networks, e.g., towards QoS-driven network slicing [31], resource allocation in cloud [43] and traffic prediction [2]. A RL algorithm designed to work over encrypted data has been proposed in [81] to enable privacy-preserving treatment of medical patients. In line of this research, we design a RL algorithm able to process data encrypted under the SSS scheme to perform VNE over a multi-domain infrastructure. To the best of our knowledge, our work is the first attempt to solve the VNE problem over encrypted data and contributes to the literature on privacy-preserving strategies for cooperative service delivery. For example, secure multiple-party computation and SSS-based approaches have been used in the context of cooperative video content delivery in [3, 6, 4].

7.3 Background

Reinforcement Learning RL is a type of machine learning technique employed to learn a model of an initially-unknown environment $\mathcal{E}$, which describes the solution space of the problem that the RL aims to solve. The solution space is represented as a set of *states* $\mathcal{S}$ which is explored by an entity referred to as *agent*. The agent moves within the environment by performing *actions* and, based on that, it receives feed-backs (i.e., *rewards*). In case of multiple environments and/or multiple agents (as in our work), the approach is often referred to as *multi-agent* RL. The objective of a RL algorithm is to learn the best action to perform according to the state in which the agent is, i.e., the action that maximizes the overall received rewards.

Q - learning Q-learning is a type of RL algorithm that models an environment as a matrix referred to as *Q-table*, whose sa-th entry represents to goodness of performing action a from state s. The model of the environment is learned by iteratively updating the Q-table as follows:

$$Q(s,a) \leftarrow Q(s,a) + lr \cdot \left(r + \gamma \cdot \max_{\hat{a}} Q(s,\hat{a}) - Q(s,a) \right) \tag{7.1}$$

where lr is the learning rate, r is the reward that the agent receives based on having performed action a from state s and γ is the discount factor.

Shamir Secret Sharing The SSS scheme [109] allows several parties to hold portions of a secret in such a way that secret reconstruction is made possible only by the cooperation of a sufficiently-large subset of them. Specifically, in a (σ, ψ) SSS scheme, the secret is divided into σ *shares* and can be reconstructed only if such subset is composed of at least ψ parties. In SSS, secret s and the corresponding set of shares $[\![s]\!]$ are defined in $\mathbb{Z}_q$, where q is a prime number greater than all the possible secrets.

Heuristics for VNE We consider two existing heuristics approaches to solve the multi-domain VNE problem, namely the Limited Information Disclosure (LID) and the Full Information Disclosure (FID). These heuristics have been proposed in [41] and are based on a relaxed linear programming formulation. LID is executed in two main phases: in the first, portions of a virtual graph are assigned to the ISPs by a centralized entity (e.g., a customer) that has a limited view of the multi-domain infrastructure. In the second, each ISP performs the optimal deployment of the received sub-graph within its network. FID is more privacy-intrusive than LID, as the centralized entity performs the optimization based on a full view of the underlying infrastructure. A deeper description of both heuristics is provided in [40].

7.4 Problem Statement

7.4.1 Problem Statement and Motivation

The formal statement of the VNE problem is the following:

$$\min \sum_{u \in \mathcal{M}} \sum_{i \in \mathcal{VN}} w_u^i d_i c_u^i x_u^i + \sum_{\substack{(i,j) \\ i \neq j}} \sum_{(u,v) \in \mathcal{L}} y_{uv}^{ij} d_{ij} c_{uv} \tag{7.2}$$

subject to:

$$\sum_{u \in \mathcal{M}} x_u^i = 1 \quad , \forall i \in \mathcal{VN} \tag{7.3}$$

$$\sum_{v \in \mathcal{M}} y_{uv}^{ij} - \sum_{v \in \mathcal{M}} y_{vu}^{ij} = x_u^i - x_u^j, \forall (i,j) \in \mathcal{VP}, \forall (u,v) \in \mathcal{L} \tag{7.4}$$

$$\sum_{i \in \mathcal{VN}} d_i x_u^i \leq \zeta_u^{(nodes)} \quad , \forall u \in \mathcal{M} \tag{7.5}$$

$$\sum_{(i,j) \in \mathcal{VP}} y_{uv}^{ij} d_{ij} \leq \zeta_{uv}^{(links)} \quad , \forall (u,v) \in \mathcal{L} \tag{7.6}$$

where $\mathcal{M}$ and $\mathcal{L}$ are the sets of physical nodes and links, respectively. u and v are the indexes of generic substrate nodes $\in \mathcal{M}$, while (u,v) indicates the link $\in \mathcal{L}$ having u and v as end-points. During the rest of the Chapter, we may indicate a generic link also as l. $\mathcal{VN}$ is the set of virtual nodes, d_i is the computational requirement of the generic VN_i, $w_u^i \in \{1, \infty\}$ is a variable indicating the feasibility of embedding VN_i into node u, c_u^i is the cost of embedding a computational unit of VN_i in node u and $x_u^i \in \{0,1\}$ is the corresponding decision variable. $\mathcal{VP}$ is the set of virtual paths, d_{ij} is the bandwidth requirement of VP_{ij}, c_{uv} is the cost of embedding a unit of bandwidth on the link connecting nodes u and v and $y_{uv}^{ij} \in \{0,1\}$ is the decision variable corresponding to the embedding of VP_{ij} in link uv. Eqs. 7.3, 7.4, 7.5, 7.6 prescribe that each VN is embedded onto exactly one physical node, the requirement of flow consistency, the fulfilment of node capacity and link capacity constraints, respectively (being $\zeta_u^{(nodes)}$ the capacity of node u and $\zeta_{uv}^{(links)}$ the capacity of link uv).

The considered problem is aimed at minimizing the overall embedding cost. Whilst this is a plausible objective for the customer, the same cannot be said for the ISPs, whose goal is the maximization of their revenues. In a multi-domain scenario, however, this optimization cannot be performed by the single ISPs, but rather by a centralized entity (such as the customer) that has an overall view of the underlying infrastructure. However, this scenario introduces privacy issues that ISPs may address, as proposed in [41], by providing only a partial view of their network to the customer. In this case (i.e., LID approach), the customer assigns portions of the virtual graph to

the ISPs, thus incurring in a *nominal embedding cost*. To protect privacy, the successive deployment of sub-graphs in ISPs' networks is performed by the ISPs themselves, and this operation introduces an additional *extra cost* that ISPs pay as a price for privacy protection. Hence, we assume the minimization of the embedding cost to be a common objective of both customer and ISPs.

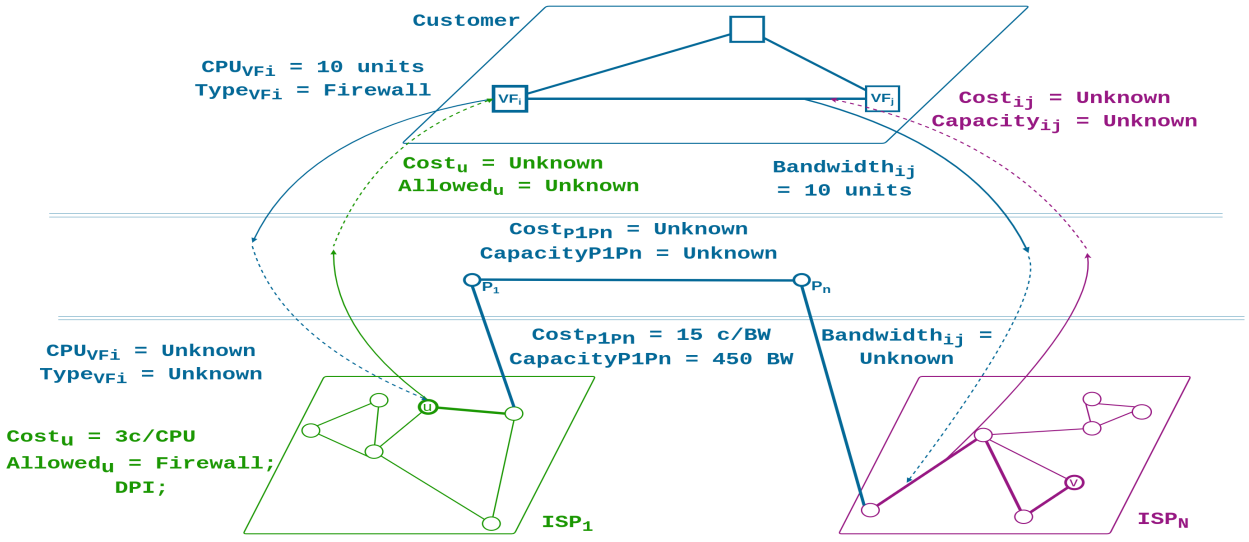


Figure 7.1: Overview of the information visible to ISPs and Customer

7.4.2 Privacy Requirements and Security Models

We consider a customer and K ISPs, whose privacy requirements are discussed in the following.

7.4.2.1 Customer

The customer aims to deploy a VG over the multi-domain infrastructure. The computational demands of the N VNs and the bandwidth requirements of the relative VPs are represented as a vector $\vec{d}$ and a matrix $\mathbf{D}$, respectively. Moreover, the types of VNs are represented as a binary matrix Δ with N rows and a number of columns equal to the number of available VNs' types (e.g., a virtual firewall).

Privacy Requirements computational demand of VN_i (i.e., d_i) and its type (i.e., δ_i) can only be disclosed to the ISP that hosts $VN_i, \forall i$ and the bandwidth requirement d_{ij} only to the ISPs that are traversed by $VP_{ij}, \forall ij$.

Security Model the customer is assumed to be an *honest-but-curious* entity that does not deviate from the licit execution of the protocol, but tries to obtain as many information as possible from the obtained data.

7.4.2.2 Internet Service Providers

The generic ISP_k owns a physical infrastructure composed of a set of $\mathcal{M}_k$ nodes which are interconnected by $\mathcal{L}_k$ links. Each node (resp., link) has a computational (resp., bandwidth) capacity, which are encoded in vectors $\zeta_k^{(nodes)}$ and $\zeta_k^{(links)}$, respectively. Moreover, the uf-th entries of matrices $\mathbf{F}_k$ and $\boldsymbol{\eta}_k$ indicate if node u can host a VN of type f and the cost of hosting it, respectively. Vector $\mathbf{C}^{(links)}$ indicates the cost of embedding a unit of bandwidth in each link.

Privacy Requirements the following information can only be known to the owner ISP: (i) capacity $\vec{\zeta}^{(nodes)}$, embedding costs $\boldsymbol{\eta}$ and feasibility $\mathbf{F}$ of the physical nodes; (ii) capacity $\vec{\zeta}^{(links)}$ and cost $\mathbf{C}^{(links)}$ of the physical links; (iii) interconnection of the internal nodes, i.e., the information if two generic nodes u, v are connected by a link. On the other hand, the interconnection of the peering nodes is assumed to be known to all the participants.

Security Model We model the ISPs as an *honest-but-curious* entity which may, for instance, be interested to obtain infrastructure details of competitor ISPs.

In Fig. 7.1, we show an overview of the information visible to the considered entities. In this figure, it is possible to identify three main layers. Going bottom up, the first layer shows the information visible only to the owner ISPs (e.g., the topology of their networks and the cost of embedding a type of VN into their nodes); the intermediate layer represents the peering interconnection, which is visible to all the participants; the top layer represents the information about the virtual graph, which are known to the customer only.

Table 7.1: Table of Notations

Variable	Description	Variable	Description
N	Number of VNs	K	Number of ISPs
$\vec{\mathbf{d}}, \mathbf{D}$	Computational (resp., bandwidth) demand of the VNs (resp., VPs)	$\mathbf{P}_k^{uv}$, $\mathcal{P}_k^{uv}$	Matrix representing the (resp., Set of) paths connecting nodes u and $v \in ISP_k$
$\mathbf{W}$	Feasibility matrix (ui-th entry is 1 if node u can embed $\in VN_i$ and ∞ otherwise)	$\mathbf{F}_k$	Matrix whose uf-th entry is 1 if node $u \in ISP_k$ can embed VN of type f (and 0 otherwise)
$\mathbf{C}_k^{(nodes)}$	Matrix of Nodes' costs (the ui-th entry is the cost of embedding a computational unit of VN_i in node $u \in ISP_k$)	$\boldsymbol{\eta}_k$	Matrix whose uf-th entry is the cost of embedding a computational unit of VN of type f in node $u \in ISP_k$
$\vec{\zeta}_k^{(nodes)}$	Computational Capacity of nodes $\in ISP_k$	$\boldsymbol{\Delta}$	VNs' types Matrix (th if-th entry is 1 if VN_i is of type f, and 0 otherwise)
$\mathcal{M}_k, \mathcal{L}_k$	Set of physical nodes (resp, links) $\in ISP_k$	$\vec{\zeta}_k^{(links)}$, $\vec{c}_k^{(links)}$	Link capacity (resp., cost) vector indicating the capacity (resp., cost of embedding a unit of bandwidth) for links $\in ISP_k$
l	Index of the generic link l	u, U	Index of the generic internal (resp., peering) node
$\mathcal{E}_{CUST}^{(i)}$, $\mathcal{S}_{CUST}^{(i)}$ $\vec{\alpha}_{CUST}^{(i)}$, $r_{CUST}^{(i)}$	Environment, state vector, action vector and reward associated with the selection of the ISP that embeds $VN_i, \forall i$	$\mathcal{E}_{CUST}^{(kk\prime,ij)}$, $\mathcal{S}_{CUST}^{(kk\prime,ij)}$ $\vec{\alpha}_{CUST}^{(kk\prime,ij)}$, $r^{(kk\prime,ij)}$	Environment, state vector, action vector and reward associated with the selection of the peering path between ISP_k and $ISP_{k\prime}$ that embeds $VP_{ij}, \forall kk\prime, ij$
$\mathcal{E}_{ISP_k}^{(i)}$, $\mathcal{S}_{ISP_k}^{(i)}$ $\vec{\alpha}_{ISP_k}^{(i)}$, $r_{ISP_k}^{(i)}$	Environment, state vector, action vector and reward associated with the selection of the ISP that embeds $VN_i, \forall i$	$\mathcal{E}_{ISP_k}^{(uv,ij)}$, $\mathcal{S}_{CUST}^{(uv,ij)}$ $\vec{\alpha}_{ISP_k}^{(uv,ij)}$, $r_{CUST}^{(uv,ij)}$	Environment, state vector, action vector and reward associated with the selection of the path between nodes u and $v \in ISP_k$ that embeds $VP_{ij}, \forall uv, ij$

7.5 The RL Algorithm for multi-domain VNE

In this Section we describe our RL-based approach to solve the VNE problem in a multi-ISPs scenario. Initially, we identify four sub-tasks in which the problem can be divided, i.e., selection of (i) the ISPs that host the VNs, (ii) the peering links that the VPs traverse, (iii) the physical nodes that embed the VNs and (iv) the intra-ISP links that embed the VPs. Note that the decisions taken in sub-task (iii) are conditioned by the output of sub-task (i). Indeed, a VN can be embedded into a physical node only if that node belongs to the infrastructure of the ISP to which the VN has been assigned in task (i). A similar reasoning can be done considering flow consistency of the VPs (i.e., Eq. 7.4).

Please notice that the solution space of the considered problem is too large to apply an exhaustive search of the optimum (i.e., the VN deployment that minimizes the total embedding cost). In such a scenario, metaheuristic approaches are commonly used. Among the possible alternatives, we employ a RL approach for the two following main reasons: (i) RL proved effective in sampling large solutions spaces efficiently and (ii) it is possible to design a privacy-preserving alternative of the Q-learning algorithm (as all the required operations can be implemented using suitable cryptographic primitives, which are presented in Section 7.6).

7.5.1 Environments

In this subsection, we define four types of RL environments that model the execution of the aforementioned sub-tasks. Each environment $\mathcal{E}$ is characterized by its state vector $\mathcal{S}$ and its $\mathcal{Q}$ matrix. $\mathcal{S}$ is a binary vector with a number of components equal to the number of states (where the only component equal to 1 is the state currently occupied by the agent), while $\mathcal{Q}$ has a row for each state and 3 columns, corresponding to the actions that the agent can perform, i.e., *left*, *stay* and *right*. As an example, if an agent is in state $[0, 1, 0, 0]$ and it performs the action right, the new state becomes $[0, 0, 1, 0]$. An action is represented as a binary vector $\vec{\alpha}$ (e.g., $\vec{\alpha} = [0, 0, 1]$ for action right) and chosen to balance between *exploitation* and *exploration*. In the former case, the action that maximizes the row of the $\mathcal{Q}$ matrix corresponding to the current state is selected (where it is assumed that indexes $0, 1, 2$ correspond to left, stay and right actions, respectively). In the latter, an action is randomly selected. The proposed environments are the following:

- An environment $\mathcal{E}^{(i)}_{CUST}$ that models the selection of the ISP in which VN_i has to be embedded, $\forall i$. $\mathcal{S}^{(i)}_{CUST}$ is a vector with K components, one for each ISP to which VN_i can be assigned.

- An environment $\mathcal{E}^{(i)}_{ISP_k}$ that models the selection of the physical node in which VN_i has to be embedded, $\forall i, k$. $\mathcal{S}^{(i)}_{ISP_k}$ is a vector with $|\mathcal{M}_k|$ components, where $\mathcal{M}_k$ is the set of physical nodes $\in ISP_k$.

- An environment $\mathcal{E}_{CUST}^{(kk\prime,ij)}$ that models the selection of the path of peering nodes connecting ISP_k and $ISP_{k\prime}$ in which VP_{ij} has to be embedded, $\forall k, k\prime, i, j$. $\mathcal{S}_{CUST}^{(kk\prime,ij)}$ is a vector with $|\mathcal{P}_{kk\prime}|$ components, where $|\mathcal{P}_{kk\prime}|$ is the number of peering paths between ISP_k and $ISP_{k\prime}$.

- An environment $\mathcal{E}_{ISP_k}^{(uv,ij)}$ that models the selection of the path (between nodes u and $v \in ISP_k$) to embed VP_{ij}, $\forall u, v, i, j$. $\mathcal{S}_{ISP_k}^{(uv,ij)}$ is a vector with $|\mathcal{P}_{uv}|$ components, one for each path that interconnects nodes u and v. As this number may be very high, practically we can consider the $|\mathcal{P}_{uv}|$ shortest paths between u and v.

We propose in Tab. 7.1 an overview of the notations used in this Chapter. Then, in the following two subsections we describe the main operations performed in the proposed RL approach. In support of the following description, we provide a pseudo-code in Algorithm 4.

7.5.2 Action Selection and State Updating

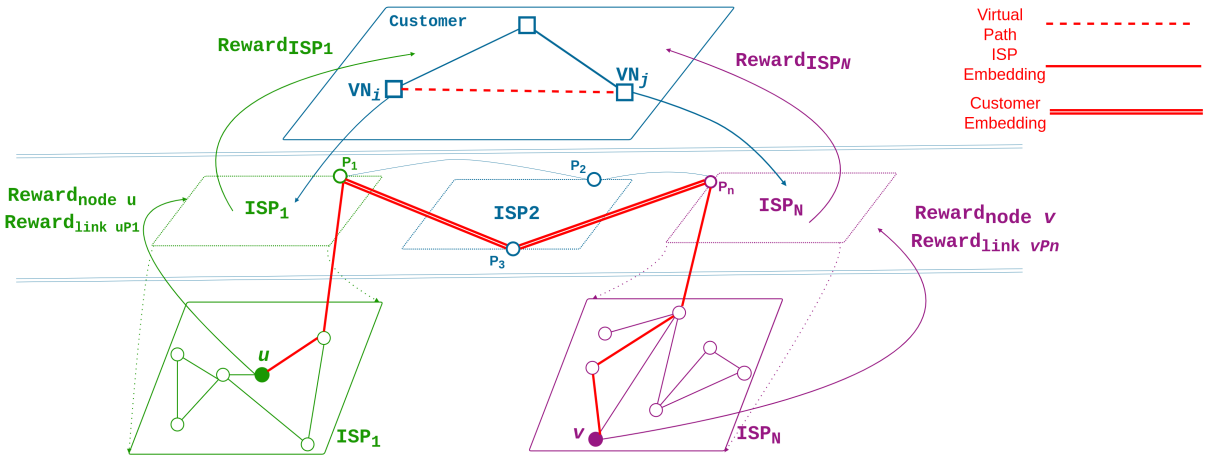


Figure 7.2: High-Level representation of the main operations performed by the proposed RL algorithm

After the initialization of the main variables (e.g., vectors states and $\mathcal{Q}$-tables), an action is performed in environment $\mathcal{E}_{CUST}^{(i)}$ and state vector $\mathcal{S}_{CUST}^{(i)}$ is changed accordingly. This operation is aimed to find the ISP in which VN_i has to be embedded, and it is repeated $\forall i$. Let us assume that VN_i has been assigned to ISP_k. An action is then performed in the environment $\mathcal{E}_{ISP_k}^{(i)}$ and the corresponding state is changed accordingly to select the physical node in which VN_i has to be embedded.

We now consider the operations relative to the selection of the peering paths on which VP_{ij} has to be embedded. Assuming that ISP_k and $ISP_{k\prime}$ are the ISPs that, at the current iteration of the RL algorithm, are required to embed VN_i and VN_j, respectively, the considered environment is

$\mathcal{E}_{CUST}^{(kk\prime,ij)}$. An action selection and successive state updating are performed in this environment to select the peering path traversed by $VP_{ij}, \forall i,j$. The selected peering path is composed of a set of peering nodes (e.g., $U_k, ..., U_{k\prime}$). As mentioned in subsection 7.4.2.2, all the involved participants are aware of the peering nodes that interconnect the ISPs (while the cost of traversing them is only known to the owner ISPs).

Once VN_i and VN_j have been assigned to their physical nodes and the peering links traversed by VP_{ij} has been chosen, operations are executed to embed VP_{ij} within the intra-ISP physical links consistently with the decisions that have been previously taken. Specifically, the flow consistency constraint of Eq. 7.4 must be fulfilled. To this end, four different scenarios must be considered: (i) ISP_k is assigned both VN_i (which is embedded in node u) and VN_j (which is embedded in node v). In this case, VP_{ij} has to be embedded within a path between nodes $u, v \in ISP_k$. To do so, action selection and state update are performed within the environment $\mathcal{E}_{ISP_k}^{(uv,ij)}$; (ii) ISP_k is assigned only VN_i (which is embedded in node u) and operations are executed in $\mathcal{E}_{ISP_k}^{(uU_k,ij)}$, where U_k is assumed to be the first peering node of the peering path that was selected in environment $\mathcal{E}_{CUST}^{(kk\prime,ij)}$; (iii) ISP_k is assigned only VN_j (which is embedded in node v) and operations are performed in the environment $\mathcal{E}_{ISP_{k\prime}}^{(U_{k\prime}v,ij)}$, where $U_{k\prime}$ is the last node of such peering path; (iv) ISP_k is assigned neither one of the two VNs and operations are performed in the environment $\mathcal{E}_{ISP_{k*}}^{(U_xU_{x+1},ij)}$, where U_x and U_{x+1} are two adjacent peering nodes of the considered peering path, which are assumed to belong to the infrastructure of ISP_{k*}.

To efficiently explore the solution space, each agent receives a reward providing a feedback on the goodness of the performed action. In our approach, actions are executed in isolated environments. To make all the agents behave towards a common optimization objective, it is crucial to carefully craft the rewards and to properly exchange them between different environments. In the next subsection, we describe the proposed reward signals, and we illustrate a high-level representation of their exchange in Fig. 7.2.

7.5.3 Rewards Computation and Q-table Updating

7.5.3.1 Embedding a VN into a physical node

The reward associated with $\mathcal{E}_{ISP_k}^{(i)}$ is defined as $r_{ISP_k}^{(i)} = -d_i \cdot c_u^i - v_u - w_u^i$, where d_i is the computational demand of VN_i and c_u^i is the cost of embedding VN_i in node u; $v_u \in \{0, \infty\}$ indicates if the node capacity constraint is fulfilled; $w_u^i \in \{1, \infty\}$ indicates if node u is eligible to host VN_i.

7.5.3.2 Embedding a VP into an intra-ISP physical path

The reward associated with $\mathcal{E}_{ISP_k}^{(uv,ij)}$ is defined as $r_{ISP_k}^{(uv,ij)} = -\sum_{l \in \mathcal{P}_{uv}^{ij}} (d_{ij} c_l + v_l) + r_{ISP_k}^{(i)} + r_{ISP_k}^{(j)}$, where l is the generic link belonging to the path $\mathcal{P}_{uv}^{ij}$ connecting nodes u and v and traversed by VP_{ij}; c_l is the cost of embedding a unit of bandwidth in link l and $v_l \in \{0, \infty\}$ is a penalty value that

Algorithm 4 Reinforcement Learning Algorithm for Multi-Domain Virtual Network Embedding

Input: $Patience$, $\mathcal{Q}^{(i)}_{CUST}, \mathcal{S}^{(i)}_{CUST}, \mathcal{Q}^{kk\prime,ij}_{CUST}, \mathcal{S}^{kk\prime,ij}_{CUST}, \mathcal{Q}^{(i)}_{ISP_k}, \mathcal{S}^{(i)}_{CUST}, \mathcal{Q}^{uv,ij}_{ISP_k}, \mathcal{S}^{uv,ij}_{ISP_k}, \forall i, \forall uv, ij, i \neq j, \forall k, \forall k\prime$

Output: $\mathcal{S}^{(i)}_{CUST}, \mathcal{S}^{kk\prime,ij}_{CUST}, \mathcal{S}^{(i)}_{CUST}, \mathcal{S}^{uv,ij}_{ISP_k}, \forall i, \forall uv, ij, i \neq j, \forall k, \forall k\prime$

Variable Initialization $Cost_{final} = \infty, epoch = 0, \quad N^{unimproved}_{epochs} = 0$
while $N^{unimproved}_{epochs} \leq Patience$ **do**
  if $epoch \equiv 0 \mod \mathcal{T}^{VF}_{CUST}$ **then**
    for $1 \leq i \leq N$ **do**
      $\vec{\alpha}^{(i)}_{CUST} \leftarrow Action\left(\mathcal{S}^{(i)}_{CUST}, \mathcal{Q}^{(i)}_{CUST}\right)$
      $\mathcal{S}^{(i)}_{CUST} \leftarrow UpdateState\left(\mathcal{S}^{(i)}_{CUST}, \vec{\alpha}^{(i)}_{CUST}\right)$
      $r^{(i)}_{CUST} \leftarrow GetReward\left(\mathcal{S}^{(i)}_{CUST}, \vec{\alpha}^{(i)}_{CUST}\right)$
    end for
  end if
  if $epoch \equiv 0 \mod \mathcal{T}^{VP}_{CUST}$ **then**
    for $1 \leq i, j \leq N, i \neq j$ **do**
      $k \leftarrow GetState\left(\mathcal{S}^{(i)}_{CUST}\right)$
      $k\prime \leftarrow GetState\left(\mathcal{S}^{(j)}_{CUST}\right)$
      $\vec{\alpha}^{VP_{kk\prime,ij}}_{CUST} \leftarrow Action\left(\mathcal{S}^{kk\prime,ij}_{CUST}, \mathcal{Q}^{(kk\prime,ij)}_{CUST}\right)$
      $\mathcal{S}^{(kk\prime,ij)}_{CUST} \leftarrow UpdateState\left(\mathcal{S}^{(kk\prime,ij)}_{CUST}, \vec{\alpha}^{(kk\prime,ij)}_{CUST}\right)$
      $r^{(kk\prime,ij)}_{CUST} \leftarrow GetReward\left(\mathcal{S}^{(kk\prime,ij)}_{CUST}, \vec{\alpha}^{(kk\prime,ij)}_{CUST}\right)$
    end for
  end if
  if $epoch \equiv 0 \mod \mathcal{T}^{VF}_{ISP}$ **then**
    for $1 \leq i \leq N$ **do**
      $k \leftarrow GetState\left(\mathcal{S}^{(i)}_{CUST}\right)$
      $\vec{\alpha}^{(i)}_{ISP_k} \leftarrow Action\left(\mathcal{S}^{(i)}_{ISP_k}, \mathcal{Q}^{(i)}_{ISP_k}\right)$
      $\mathcal{S}^{(i)}_{ISP_k} \leftarrow UpdateState\left(\mathcal{S}^{(i)}_{ISP_k}, \vec{\alpha}^{(i)}_{ISP_k}\right)$
      $r^{(i)}_{ISP_k} \leftarrow GetReward\left(\mathcal{S}^{(i)}_{ISP_k}, \vec{\alpha}^{(i)}_{ISP_k}\right)$
    end for
  end if
  for $1 \leq i, j \leq N, i \neq j$ **do**
    $k \leftarrow GetState\left(\mathcal{S}^{(i)}_{CUST}\right)$
    $k\prime \leftarrow GetState\left(\mathcal{S}^{(j)}_{CUST}\right)$
    $Path^{(kk\prime,ij)}_{peering} \leftarrow GetPeeringPath\left(\mathcal{S}^{(kk\prime,ij)}_{ISP_{CUST}}\right)$
    for $Link_{peering} \in Path_{peering}$ **do**
      $u, v \leftarrow EndPoints\left(Link_{peering}\right)$
      $k \leftarrow OwnerISP\left(u, v\right)$
      $\vec{\alpha}^{(uv,ij)}_{ISP_k} \leftarrow Action\left(\mathcal{S}^{(uv,ij)}_{ISP_k}, \mathcal{Q}^{(uv,ij)}_{CUST}\right)$
      $S^{(uv,ij)}_{ISP_k} \leftarrow UpdateState\left(\mathcal{S}^{(uv,ij)}_{ISP_k}, \vec{\alpha}^{(uv,ij)}_{ISP_k}\right)$
      $r^{(uv,ij)}_{ISP_k} \leftarrow GetReward\left(\mathcal{S}^{(uv,ij)}_{ISP_k}, \vec{alpha}^{(uv,ij)}_{ISP_k}\right)$
      $\mathcal{Q}^{(uv,ij)}_{ISP_k} \leftarrow UpdateQ\left(\mathcal{Q}^{(uv,ij)}_{ISP_k}, \vec{alpha}^{(uv,ij)}_{ISP_k}\right)$
    end for
  end for
  if $epoch \geq 1 \;\&\; epoch \equiv 0 \mod \mathcal{T}^{VF}_{ISP} - 1$ **then**
    for $1 \leq i \leq N$ **do**
      $k \leftarrow GetState\left(\mathcal{S}^{(i)}_{CUST}\right)$
      $r^{(i)}_{ISP_k} \leftarrow r^{(i)}_{ISP_k} + \sum_{u,v} r^{(uv,ij)}_{ISP_k}$
      $\mathcal{Q}^{(i)}_{ISP_k} \leftarrow UpdateQ\left(\mathcal{Q}^{(i)}_{ISP_k}, \vec{alpha}^{(i)}_{ISP_k}\right)$
    end for
  end if
  if $epoch \geq 1 \;\&\; epoch \equiv 0 \mod \mathcal{T}^{VP}_{CUST} - 1$ **then**
    for $1 \leq i, j \leq N, i \neq j$ **do**
      $k \leftarrow GetState\left(\mathcal{S}^{(i)}_{CUST}\right)$
      $k\prime \leftarrow GetState\left(\mathcal{S}^{(j)}_{CUST}\right)$
      $r^{(kk\prime,ij)}_{CUST} \leftarrow r^{(kk\prime,ij)}_{CUST} + \sum_k \sum_{u,v} r^{(uv,ij)}_{ISP_k}$
      $\mathcal{Q}^{(kk\prime,ij)}_{CUST} \leftarrow UpdateQ\left(\mathcal{Q}^{(kk\prime,ij)}_{CUST}, r^{(kk\prime,ij)}_{CUST}\right)$
    end for
  end if
  if $epoch \geq 1 \;\&\; epoch \equiv 0 \mod \mathcal{T}^{VN}_{CUST} - 1$ **then**
    for $1 \leq i \leq N$ **do**
      $\mathcal{Q}^{(i)}_{CUST} \leftarrow UpdateQ\left(\mathcal{Q}^{(i)}_{CUST}, r^{(i)}_{CUST}\right)$
    end for
  end if
  $Current_{cost} \leftarrow ComputeCost\left(\mathcal{S}_{CUST}, \mathcal{S}_{ISP_k} \forall k\right)$
  if $Current_{cost} \leq Final_{cost}$ **then**
    $Final_{Cost} = Current_{cost}$
    $N^{unimproved}_{epochs} = 0$
  else
    $N^{unimproved}_{epochs} \leftarrow N^{unimproved}_{epochs} + 1$
  end if
  $epoch \leftarrow epoch + 1$
  if $N^{unimproved}_{epochs} \geq Patience$ **then**
    break
  end if
end while

return $\mathcal{S}^{(i)}_{CUST}, \mathcal{S}^{kk\prime,ij}_{CUST}, \mathcal{S}^{(i)}_{CUST}, \mathcal{S}^{uv,ij}_{ISP_k}, \forall i, \forall uv, ij, i \neq j, \forall k, \forall k\prime$

describes the fulfilment of the link capacity constraint. $r^{(i)}_{ISP_k}$ and $r^{(j)}_{ISP_k}$ are the rewards associated with the embedding of VN_i and VN_j (i.e., the end-points of the considered VP), which have been described in the previous subsection.

7.5.3.3 Assigning a VN to an ISP

The reward associated with $\mathcal{E}^{(i)}_{CUST}$ is defined as $r^{(i)}_{CUST} = \sum_{k=1}^{K} \sum_{j=1}^{N} r^{(j)}_{ISP_k}$. This reward is the same for all the VNs, i.e., $\forall i$, to provide a feedback that considers the current embedding of all the VNs.

7.5.3.4 Assigning a VP to a peering path

The reward associated with $\mathcal{E}^{(kk\prime,ij)}_{CUST}$ is defined as $r^{(kk\prime,ij)}_{CUST} = \sum_{(u,v)} \sum_{k=1}^{K} r^{uv,ij}_{ISP_k}$. This reward is the summation of the single rewards relative to the embedding of the VP_{ij} within all the K ISPs.

Once the reward relative to an environment has been computed, the corresponding $\mathcal{Q}$-table is updated according to Eq. 7.1. Since the considered sub-tasks can be solved at different time scales (e.g., the selection of the ISP in which a VN is embedded can be performed less frequently than the selection of the embedding node), we define the periods of execution of such sub-tasks (measured in number of RL iteration) as $\mathcal{T}^{VN}_{CUST}, \mathcal{T}^{VP}_{CUST}, \mathcal{T}^{VN}_{ISP}, \mathcal{T}^{VP}_{ISP}$. At every iteration of the RL algorithm the current embedding cost is computed and all the described operations are repeated until no improvement to the embedding cost is observed for a number of iterations equal to *patience*.

As it may be noticed, we design rewards that are strictly related to the embedding costs. Therefore, from their exchange, the participants obtain information about other parties' data that, if properly analyzed, can be used to violate the privacy requirements in subsection 7.4.2. To address this issue, we propose a privacy-preserving version of the RL approach, which is build on the elements that we describe in the following Section.

7.6 Building Blocks for Privacy-Preserving RL

7.6.1 Representation of Data Suitable for Secure Computation

As presented in subsection 7.4.2, data owned by the customer and the ISPs are arranged in vector/-matrix form (e.g., vector $\vec{d}$ and matrix $\mathcal{D}$ to represent computational and bandwidth requirements). Each element of these vectors and matrices can be distributed among the participants as a set of shares, thus allowing secure computation on them. In addition to these data, we also a propose a representation of topological relations among the networks' nodes suitable to perform secure computations. Specifically, we represent the paths connecting two generic nodes u and v as a matrix $\mathbf{P}_{uv}$, with a number of rows equal to the number of paths connecting nodes u and v (i.e., $|\mathcal{P}^{uv}_k|$) and a number of columns equal to the total number of links of the infrastructure (i.e., $|\mathcal{L}_k|$). Each path can be represented as a binary vector, whose l-th element is 1 if the path contains the l-th link of the ISP's infrastructure. An example of this representation is shown in the following:

Table 7.2: Data Overhead

Operation	Bits exchanged between each pair of parties	
Name	Online	Offline
Random	0	B
Mult	B	$4B$
MultDec	$(m+1) \cdot B$	$m \cdot B$
EQ	B^2	$6B^2$
GE	$9B^2$	0

$$
\begin{array}{c} \\ Path_1 \\ Path_2 \\ \cdots \\ Path_{|\mathcal{P}_k^{uv}|} \end{array}
\begin{array}{c}
\begin{array}{ccccc} Link_1 & \cdots & Link_l & \cdots & Link_{|\mathcal{L}_k|} \end{array} \\
\begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}
\end{array}
$$

and it is defined for each possible pair of nodes, i.e., $\forall u, v, u \neq v$.

7.6.2 Existing Privacy-Preserving Primitives

In this Subsection, we describe several existing operators, and we show in Table 7.2 the amount of data that each pair of participants need to exchange to execute them. By *on-line*, we refer to a data exchange that must be done contextually to the execution of the operation and cannot be performed in advance (as in the *off-line* case).

7.6.2.1 Secure generation of the shares of a random number

By executing **Random**, a share $[\![rv]\!]$ of a random variable is learnt by each participant (none of which knows the secret rv). We employ the implementation described in [112].

7.6.2.2 Secure multiplication

Mult takes in input the shares $[\![x]\!], [\![y]\!]$ and returns $[\![z]\!]$, where $z = x \cdot y$. We employ the protocol presented in [20].

7.6.2.3 Secure multiplication with a decimal number

MultDec takes in input a share $[\![x]\!]$ and a plain decimal value λ, and returns the share of their product $[\![\lambda \cdot x]\!]$. Exploiting the fact that a decimal number can be represented as the ratio between properly chosen integer numerator λ_{num} and denominator λ_{den}, the operations performed by this subroutine can be split into two main parts: the first is the multiplication by λ_{num} (which can be

easily performed as $[\![\lambda \cdot x]\!] = \lambda \cdot [\![x]\!]$) and the second is the division by λ_{den}. To this end, we employ the protocol presented in [27], which takes as input a secret shared integer $[\![x]\!]$ and an integer m and returns $[\![\lfloor \frac{x}{2^m} \rfloor + \rho]\!]$, where $\rho \in \{0, 1\}$ is a random variable encoding the rounding strategy.

7.6.2.4 Secure equality test and greater-or-equal

EQ (resp., **GE**) takes in input the shares $[\![x]\!]$ and $[\![y]\!]$ and returns the share $[\![b_{eq}]\!]$ (resp., $[\![b_{ge}]\!]$), where $b_{eq} = 1$ (resp., $b_{ge} = 1$) iff $x = y$ (resp., $x \geq y$) and 0 otherwise. In this study, we employ the implementations of EQ and GE described in [112].

All the privacy-preserving operators described until now are well-known and publicly available. However, they are not sufficient to build our privacy-preserving RL algorithm, for which additional operators are needed. We develope such novel operators based on the aforementioned existing privacy-preserving primitives, and we describe them in the following subsection.

7.6.3 New Privacy-Preserving Operators

7.6.3.1 Secure computation of the maximum element of a vector and corresponding index

Max (resp., **ArgMax**) takes as input a vector of shares $\vec{x} = [[\![x_1]\!], [\![x_2]\!], ..., [\![x_\Phi]\!]]$ and returns $[\![\max(\vec{x})]\!]$ (resp., $[\![i^*]\!] = [\![\arg\max_i(\vec{x})]\!]$). These operators are described in Algorithm 5.

Algorithm 5 Computing the maximum value (along with corresponding index) of a vector of shared secret values

Input: $[\![\vec{x}]\!] = [[\![x_1]\!], [\![x_2]\!], ..., [\![x_\Phi]\!]]$
Output: $[\![\max(\vec{x})]\!], [\![\beta^*]\!] = [\![\arg\max_\beta(\vec{x})]\!]$

1: Delivery of $[\![1]\!]$ from a given party to all the others
2: Initialize $\mathbf{max} = [\![x_1]\!]$
3: Initialize $\mathbf{argmax} = [\![1]\!]$
4: **for** $\beta \in \{2, ..., \Phi\}$ **do**
5: $\quad [\![b_\beta^{(ge)}]\!] \leftarrow$ Apply the **GE** operator on $(\mathbf{max}, [\![x_\beta]\!])$
6: $\quad \mathbf{argmax} \leftarrow \mathbf{argmax} \cdot \left(1 - [\![b_\beta^{(ge)}]\!]\right) + [\![b_\beta^{(ge)}]\!] \cdot \beta$
7: $\quad \mathbf{max} \leftarrow \mathbf{max} \cdot \left(1 - [\![b_\beta^{(ge)}]\!]\right) + [\![b_\beta^{(ge)}]\!] \cdot [\![x_\beta]\!]$
8: **end for**

return $\mathbf{max} = [\![\max(\vec{x})]\!], \mathbf{argmax} = [\![\arg\max_\beta(\vec{x})]\!]$

7.6.3.2 Secure update of the states vector

UpdateState takes as input the current vector state $\vec{s} = [[\![s_1]\!], [\![s_2]\!], ..., [\![s_\Phi]\!]]$ and outputs the new state $\vec{s\prime}$, according to the action executed by the agent (e.g., $[\![left]\!] = [\![1]\!]$, $[\![stay]\!] = [\![0]\!]$ and $[\![right]\!]$

$= [\![0]\!]$ if the agent chooses the action left). The β-th component of the updated state is derived as $[\![s'_\beta]\!] = [\![s_{\beta-1}]\!] \cdot [\![right]\!] + [\![s_\beta]\!] \cdot [\![stay]\!] + [\![s_{\beta+1}]\!] \cdot [\![left]\!]$.

Masked Secure Update of the states vector The **MaskedUpdateState** may be employed in case the participants are not aware of the vector state to modify and only know $[\![mask]\!]$, where the binary value $mask$ is 1 if the considered state has to be updated. This subroutine updates the output of UpdateState as $[\![s'_\beta]\!] \leftarrow [\![mask]\!] \cdot [\![s'_\beta]\!] + (1 - [\![mask]\!]) \cdot [\![s_\beta]\!]$.

7.6.3.3 Secure Selection of the row of a matrix

RowSelection takes as input a matrix $\vec{\mathcal{M}} \in \mathbb{Z}_q^{\Phi X \Omega}$ and a vector $\vec{x} = [[\![x_1]\!], [\![x_2]\!], ..., [\![x_\Phi]\!]]$. All the components of $\vec{x}$ are $[\![0]\!]$, except the one corresponding to the row to select that is $[\![1]\!]$ (say $[\![x_{\beta*}]\!] = [\![1]\!]$). RowSelection returns a vector $\vec{\mathcal{M}}_{\beta*}$ corresponding to the selected row. The χ-th component of this vector is given by $[\![\mathcal{M}_{\beta*}(\chi)]\!] = \sum_{\beta=1}^{\Phi} [\![\vec{\mathcal{M}}(\beta, \chi)]\!] \cdot [\![x_\beta]\!]$.

7.6.3.4 Secure Action Selection

SelectAction takes as input a $\mathcal{Q}$-table matrix $\mathcal{Q} \in \mathbb{Z}_q^{\Phi X 3}$, a decimal number $\upsilon \in [0, 1]$ and a vector representing the current state of an agent (i.e., $\vec{s} = [[\![s_1]\!], [\![s_2]\!], ..., [\![s_\Phi]\!]]$). This subroutine returns the action α that the agent should perform as $[[\![left]\!], [\![stay]\!], [\![right]\!]]$ (where only one component is $[\![1]\!]$ and the others are $[\![0]\!]$).

The approach to follow, i.e., exploitation or exploration, is chosen according to the realization of a binary random variable, which returns exploitation with probability υ. For simplicity, we mandate one of the participants to obtain such value and to communicate it to all the others. In case of exploitation, RowSelection is employed to obtain the row of the $\mathcal{Q}$-table associated with the current state encoded in $\vec{s}$. Then, ArgMax is used to compute $[\![\arg\max_\chi \mathcal{Q}(s)]\!]$, which is needed to compute $[\![action]\!]$ using the EQ operator for all the three possible actions. For example, $[\![action]\!] = EQ([\![\arg\max_\chi \mathcal{Q}(s)]\!], [\![2]\!])$ for $action = right$ and derive the action vector $\vec{\alpha} = [[\![left]\!], [\![stay]\!], [\![right]\!]]$. In case of exploration, the Random subroutine is executed to generate the shares of a random value, i.e., $[\![RV]\!] \in [0, q-1]$. Then, the GE operator is used to perform $[\![left]\!] = 1 - GE([\![rv]\!], [\![\frac{q}{3}]\!]), [\![stay]\!] = GE([\![rv]\!], [\![\frac{q}{3}]\!]) \cdot \left(1 - GE([\![rv]\!], [\![\frac{2q}{3}]\!])\right), [\![right]\!] = GE([\![RV]\!], [\![\frac{2q}{3}]\!])$ and derive the action vector $\vec{\alpha}$.

7.6.3.5 Secure Computation of VN Embedding Cost

CostEmbeddingVN takes as input the state vector $S_{ISP_k}^{(i)}$, the cost vector $\vec{c}^{(nodes)}$ and the computational demand of VN_i, i.e., $[\![d_i]\!]$ and returns $[\![d_i \cdot c_u^{(nodes)}]\!]$ corresponding to the u-th node in which VN_i is embedded. An element-wise secure multiplication of vectors $S_{ISP_k}^{(i)}$ and $\vec{c}^{(nodes)}$ is executed.

The obtained products are then summed up, and the result is securely multiplied with $[\![d_i]\!]$ to get $[\![d_i \cdot c_u^{(nodes)}]\!]$.

7.6.3.6 Secure Computation of VP Embedding Cost

CostEmbeddingVP takes as input the state vector $\mathcal{S}_{ISP_k}^{uv,ij}$, which encodes the physical path connecting nodes u and v (belonging to ISP_k) that is currently traversed by VP_{ij}, the matrix $\mathbf{P}_k^{uv}$, which encodes all the paths connecting nodes u and v, as described in Subsection 7.6.1, the cost of traversing the links of ISP_k, i.e., $\vec{c}_k^{(links)}$ and the amount of traffic exchaged between VN_i and VN_j, i.e., $[\![d_{ij}]\!]$. The subroutine returns $\sum_{l \in \mathcal{P}_{uv}^{ij}} [\![d_{ij} \cdot c_l^{links}]\!]$, i.e., the cost of embedding VP_{ij} in its current physical path. The RowSelection operator is applied to matrix $\mathbf{P}_k^{uv}$ and to state vector $\mathcal{S}_{ISP_k}^{uv,ij}$ to select the path belonging to ISP_k traversed by VP_{ij}. Then, Mult is used to perform a secure element-wise multiplication of $\vec{c}^{(nodes)}$ and the selected row. All the elements of the obtained vector are summed up and the result is securely multiplied by $[\![d_{ij}]\!]$ using the Mult subroutine to obtain $\sum_{l \in \mathcal{P}_{uv}^{ij}} [\![d_{ij} \cdot c_l^{links}]\!]$.

7.6.3.7 Secure Node's Embedding Feasibility and Cost

NodeFeasibility takes as input the matrices $\vec{F}$ and $\vec{\Delta}$. The uf-th element of $\vec{F}$ and the if-th element of $\vec{\Delta}$ are $[\![1]\!]$ if a VN of type f can be hosted in node u and if VN_i is of type f, (and $[\![0]\!]$ otherwise). This subroutine returns a matrix $\vec{W}$, whose ui-th element is $[\![0]\!]$ in case VN_i can be hosted in node u and $[\![\infty]\!]$[1] otherwise. The u-th row of matrix $\vec{F}$ and the i-th row of matrix $\vec{\Delta}$ are multiplied element-wise using the Mult operator. Resulting products are then summed up to obtain $[\![w_u^i]\!]$, which is $[\![1]\!]$ in case node u is eligible to host VN_i (and $[\![0]\!]$ otherwise). $[\![w_u^i]\!]$ is then updated as $[\![w_u^i]\!] \leftarrow (1 - [\![w_u^i]\!]) \cdot \infty + [\![w_u^i]\!]$

Similarly, the subroutine **NodeCost** takes as input the matrices $\boldsymbol{\eta}$ and $\boldsymbol{\Delta}$, where the uf-th element of $\boldsymbol{\eta}$ is the cost of embedding the VN of type f in u. This subroutine returns a matrix $\mathbf{C}$, whose ui-th element is the cost of embedding a computational unit of VN_i in node u. The u-th row of matrix $\boldsymbol{\eta}$ and the i-th row of matrix $\boldsymbol{\Delta}$ are multiplied element-wise using the Mult operator and the resulting products are then summed up to provide the cost of embedding VN_i in u.

7.6.3.8 Secure Node's Capacity Constraint Verification

NodeCapacity takes as input the states vector $\mathcal{S}_{CUST}^{(i)}$ and $\mathcal{S}_{ISP_k}^{(i)}, \forall i$, the VNs' computational demand vector $\vec{d}$, the nodes' capacity vector $\vec{\zeta}^{(nodes)}$ and the index u of a physical node $\in ISP_k$. The output $v_u \in \{0, \infty\}$ indicates the fulfilment of the capacity constraint for node u. The k-th element of $\mathcal{S}_{CUST}^{(i)}$ (which is $[\![1]\!]$ iff VN_i has been assigned to ISP_k) is securely multiplied with $[\![d_i]\!]$ and $[\![\mathcal{S}_{ISP_k}^{(i)}]\!]_u$ (which is $[\![1]\!]$ iff VN_i is embedded in node u) by recursively applying the Mult operator.

[1] ∞ is encoded with the value 1000 in the performed experiments

This operation is repeated $\forall i$, and the results are summed up to obtain the current amount of computational demand embedded in node u. This value is then securely compared with the capacity of the node $[\![\zeta_u^{(nodes)}]\!]$ using the GE operator. The result of this comparison is successively multiplied by ∞ to obtain v_u (which is equal to 0 if node's capacity is not exceeded, and ∞ otherwise).

7.6.3.9 Secure Links' Capacity Constraint Verification

LinkCapacity takes as input the states vector $\mathcal{S}_{ISP_k}^{(uv,ij)}, \forall i,j,i \neq j, \forall u,v$ and a matrix $\mathbf{P}_k^{uv}$ representing the paths connecting nodes u and v (as described in Subsection 7.6.1), the value $[\![d_{ij}]\!]$ and the index l. This subroutine returns $[\![v_l]\!]$, where $v_l \in \{0, \infty\}$ indicates the fulfilment of the capacity constraint for link l. To select the path connecting nodes u, v that is traversed by VP_{ij}, RowSelection is applied on $\mathbf{P}_k^{uv}$. Then, Mult is executed to perform the secure multiplication between $[\![d_{ij}]\!]$ and the l-th element of the selected vector. This operation is repeated $\forall u, v, \forall i, j, i \neq j$ and the results are summed up to obtain the share of the amount of bandwidth that are currently deployed on the l-th link. Finally, the GE operator is applied to securely compare this value and $\vec{\zeta}_l^{(links)}$. The result of this operation is then multiplied by ∞ to obtain $[\![v_l]\!]$, which is ($[\![0]\!]$ in case the capacity of the l-th link is not exceeded).

7.6.3.10 Secure Updating of the Q-table

UpdateQ inputs the following data: a state vector $\vec{s} = [[\![s_1]\!], ..., [\![s_\Phi]\!]]$, an action vector $\vec{\alpha}$, a $\mathcal{Q}$-table $\in \mathbb{Z}^{\Phi X 3}$, a reward $[\![r]\!]$ and two decimal values, i.e., the learning rate lr and the discount factor γ. The output of this subroutine is the updated $\mathcal{Q}$-table, which is equal to the matrix $\mathcal{Q}$ in input, except for the s, χ-th entry (which corresponds to the selected action in the current state of the agent), which is modified according to Eq. 7.1, that we repropose for clarity of exposition: $\mathcal{Q}(s,\chi) \leftarrow \mathcal{Q}(s,\chi) + lr \cdot (r + \gamma \cdot \max_{\hat{\chi}} \mathcal{Q}(s,\hat{\chi}) - \mathcal{Q}(s,\chi))$.

Initially, RowSelection is employed on $\mathcal{Q}$ and $\vec{s}$ to obtain the row corresponding to the current state. Max is then used to compute the maximum value of this row, i.e., $[\![\max_{\hat{\chi}} Q(s,\hat{\chi})]\!]$, which is then multiplied by the discount factor γ using the MultDec subroutine. The values within parenthesis are successively summed up and multiplied by lr, also using the MultDec subroutine. At this point, the obtained value need to be summed to $\mathcal{Q}(s,\chi)$ only, while all the other values of the matrix must remain unchanged. As participants are not aware of s and χ, all the elements of the matrix $\mathcal{Q}$ must be summed to the value $lr \cdot (r + \gamma \cdot \max_{\hat{\chi}} \mathcal{Q}(s,\hat{\chi}) - \mathcal{Q}(s,\chi))$ multiplied by a properly selected $[\![mask]\!]$, which can be obtained as follows: the Mult operator is applied to $[\![\vec{s}(\phi)]\!]$ and $[\![\vec{\alpha}(\chi)]\!], \forall \phi, \chi$, in such a way that $[\![mask]\!]$ is $[\![1]\!]$ only for the $\hat{s}\hat{\chi}$-th entry (i.e., that corresponding of the current state and selected action), and $[\![0]\!]$ otherwise.

7.6.3.11 Masked Secure Updating of the $\mathcal{Q}$-table

MaskedUpdateQ differs from UpdateQ as the value $(r + \gamma \cdot \max_{\hat{\chi}} \mathcal{Q}(s, \hat{\chi}) - \mathcal{Q}(s, \chi))$ is securely multiplied by $[\![mask]\!]$ before being multiplied by lr using MultDec.

7.7 Privacy-Preserving RL for VNE

In this Section, we describe the privacy-preserving version of the algorithm described in Section 7.5, that performs operations on the shares exchanged between customer and ISPs. We consider a $(K+1, K+1)$ SSS, i.e., the secrets can be reconstructed only if all the ISPs and the customer cooperate. Each participant generate $K+1$ shares of its secrets, among which one is taken for itself and the remaining delivered to the other parties.

7.7.1 Initial Data Sharing

7.7.1.1 Secret Sharing of the Data between customers and ISPs

Initially, the participants exchange with each other the following data:

Shares Distributed by the customer The customer distributes to the ISPs, in secret shared form, the data described in subsection 7.4.2.1, i.e., the vector of computational demands $\vec{d}$, the feasibility matrix $\vec{\Delta}$ and the bandwidth demand matrix $\vec{D}$. Moreover, it also distributes $\mathcal{Q}^{(i)}_{CUST}$ and $\mathcal{S}^{(i)}_{CUST}, \forall i$ and $\mathcal{Q}^{(kk\prime,ij)}_{CUST}$ and $\mathcal{S}^{(kk\prime,ij)}_{CUST}, \forall k, k\prime, \forall i, j, i \neq j$.

Shares Distributed by the ISPs Each ISP distributes, in secret-shared form, the following data: (i) the nodes' computational capacity vector $\vec{\zeta}^{(nodes)}$, (ii) a feasibility matrix $\vec{\mathbf{F}}$ indicating the types of VNs that can be hosted in its physical nodes, (iii) the nodes' embedding cost matrix $\boldsymbol{\eta}$, the link capacity vector $\vec{\zeta}^{(links)}$, (iv) $\mathcal{Q}^{(i)}_{ISP_k}$ and $\mathcal{S}^{(i)}_{ISP_k}, \forall i$; the (iv) $\mathcal{Q}^{(uv,ij)}_{ISP_k}$ and $\mathcal{S}^{(uv,ij)}_{ISP_k}, \forall i, j, i \neq j, \forall u, v$.

Initially, the NodeFeasibility and NodeCost subroutines are applied to $\boldsymbol{\Delta}$,$\mathbf{F}$ and $\boldsymbol{\Delta}$,$\boldsymbol{\eta}$, respectively, to obtain the information on the feasibility and cost of embedding the VNs on the physical nodes, i.e., $\mathbf{W}$ and $\mathbf{C}$. With these data in hands, participants can perform the privacy-preserving counterparts of the operations described in Algorithm 4, which are described in the following subsection. Note that, as explained in Section 7.5, the operations performed within an environment may depend on the operations performed in another one (e.g., the placement of a VN within the network of an ISP is consequent to the selection of ISP hosting it). When performing operations on secret shares, however, the participants are not aware of the decisions taken and, consequently, they do not know the environments they should act in (i.e., they do not know which state vector and relative $\mathcal{Q}$-table has to be updated). To address this issue, we associate a $[\![mask]\!]$ with each considered environment, where $mask \in \{0, 1\}$ encodes the information on the goodness of performing operations

on it. Whenever needed, we will explain the procedures followed by the participants to obtain these masks.

7.7.2 Privacy-Preserving Operations on the Environments

7.7.2.1 Operations on $\mathcal{E}^{(i)}_{ISP_k}$

In the following we describe the privacy-preserving counterpart of the operations presented in lines 19 : 25 and 41 : 45 of Algorithm 4, which aim to find the physical node in which VN_i has to be embedded. Firstly, the action $\vec{\alpha}^{(i)}_{ISP_k}$ is obtained using the SelectAction subroutine. The state $\mathcal{S}^{(i)}_{ISP_k}$ is then updated accordingly by means of the MaskedUpdateState subroutine, where the employed mask is the k-th component of $\mathcal{S}^{(i)}_{CUST}$, which is $[\![1]\!]$ iff VN_i has been assigned to ISP_k. The corresponding reward is $r^{(i)}_{ISP_k} = [\![-d_i \cdot c^{(i)}_u - v_u - w^i_u]\!]$, which is obtained using the CostEmbeddingVN, NodeCapacity and NodeFeasibility operators. Then, $\mathcal{Q}^{(i)}_{ISP_k}$ is updated using the MaskedQUpdate subroutine. These operations are repeated $\forall i, k$.

7.7.2.2 Operations on $\mathcal{E}^{uv,ij}_{ISP_k}$

Here we describe the privacy-preserving counterpart of the operations presented in lines 27 : 38 of Algorithm 4 to select the path connecting nodes u and v in which VP_{ij} has to be embedded. We remind that the selection of the physical path connecting nodes u and v belonging to ISP_k on which VP_{ij} should pass is dependent on the peering path that such VP traverses. Since customer and ISPs are not aware of the peering path in which VP_{ij} is embedded, they compute $[\![mask]\!] = [\![(VN_i \in ISP_k) \cdot (VN_j \in ISP_{k\prime}) \cdot \mathcal{S}^{(ij,kk\prime)}_{CUST}[\beta]]\!]$, i.e., $mask = 1$ iff VN_i and VN_j have been assigned to ISP_k and $ISP_{k\prime}$ and VP_{ij} is embedded in the β-th peering path connecting them ($mask = 0$ otherwise).

Then, the participants consider the environment $\mathcal{E}^{uU_k,ij}_{ISP_k}$, where U_k is assumed to be the first peering node of the β-th peering path. SelectAction is executed to obtain $\vec{\alpha}^{uU_k,ij}_{ISP_k}$, which is successively used to update the corresponding state vector by means of the MaskedUpdateState subroutine. At this point, the reward $[\![r^{(uv,ij)}_{ISP_k}]\!] = -\sum_{l \in \mathcal{P}^{ij}_{uv}} ([\![c_l \cdot d_{ij}]\!] + [\![v_l]\!]) + [\![r^{(i)}_{ISP_k}]\!] + [\![r^{(j)}_{ISP_k}]\!]$ is computed using the CostEmbeddingVP and the LinkCapacity operator to obtain $\sum_{l \in \mathcal{P}^{ij}_{uv}} [\![c_l \cdot d_{ij} + v_l]\!]$, and by summing the shares obtained as described in Subsection 7.7.2.1 to obtain $[\![r^{(i)}_{ISP_k} + r^{(j)}_{ISP_k}]\!]$. With these values in hand, $\mathcal{Q}^{uv,ij}_{ISP_k}$ is updated with the MaskedQUpdate operator. The same process in performed considering the last node of the peering path (say $U_{k\prime}$) and the environment $\mathcal{S}^{U_{k\prime}v,ij}_{ISP_{k\prime}}, \forall v$ and the intermediate peering nodes (say U_{x_i} and $U_{x_{i+1}}$), for which the same operations are performed on the environment $\mathcal{E}^{U_{x_i}U_{x_{i+1}},ij}_{ISP_{k\prime}}$. All these operations are repeated $\forall u, v, i, j, k$.

7.7.2.3 Operations on $\mathcal{E}^i_{CUST}$

Here we describe the privacy-preserving counterpart of the operations presented in lines 4 : 8 and 56 : 58 of Algorithm 4 to select the ISP in which VN_i has to be embedded. Firstly, action vector $\vec{\alpha}^{(i)}_{CUST}$ is obtained by means of the SelectAction subroutine and used to update the state vector $\mathcal{S}^{(i)}_{CUST}$ employing the UpdateState subroutine. Then, the reward is computed as $r^{(i)}_{CUST} = \sum_{k=1}^{K}\sum_{j=1}^{N}[\![r^{(j)}_{ISP_k}]\!]$, where $[\![r^{(j)}_{ISP_k}]\!]$ is obtained as described in Subsection 7.7.2.1. With these data, $\mathcal{Q}^i_{CUST}$ is updated using the UpdateQ operator. These operations are repeated $\forall i$.

7.7.2.4 Operations on $\mathcal{E}^{(kk\prime,ij)}_{CUST}$

Here we describe the privacy-preserving counterpart of the operations presented in lines 11 : 17 and 48 : 52 of Algorithm 4 to select the peering path in which VP_{ij} has to be embedded. Action $\vec{\alpha}^{(kk\prime,ij)}_{CUST}$ is obtained using the SelectAction subroutine and used to update the vector state $\mathcal{S}^{kk\prime,ij}_{CUST}$ using the MaskedUpdateState subroutine. The considered mask is obtained applying the Mult operator on $[\![VN_i \in ISP_k]\!]$ and $[\![VN_j \in ISP_{k\prime}]\!]$. Then, the reward $r^{kk\prime,ij}_{CUST} = \sum_{(u,v)}\sum_{k=1}^{K}[\![r^{uv,ij}_{ISP_k}]\!]$ is obtained by summing the single rewards computed as explained in Subsection 7.7.2.2. Finally, $\mathcal{Q}^{kk\prime,ij}_{CUST}$ is updated using the MaskedQUpdate operator. These operations are repeated $\forall kk\prime, ij$.

7.7.3 Computation of the Embedding Cost

At each iteration of the RL algorithm the participants know $[\![d_i \cdot c^i_u]\!], \forall u, i, [\![v_u]\!], \forall u,\ [\![w^i_u]\!], \forall u, i$, which are obtained during the computation of the rewards relative to the embedding of the VNs into the physical nodes, as explained in subsection 7.7.2.1. Similarly, the participants also obtain $[\![\sum_{l\in\mathcal{P}^{ij}_{uv}} d_{ij}c^{links}_l]\!], \forall ij, l$ and $[\![v_l]\!], \forall l$ from the computation of the rewards corresponding to the embedding of the VPs into the physical links, as explained in subsection 7.7.2.2.

By computing $\sum_i [\![d_i \cdot c^i_u]\!] + \sum_{ij}[\![d_{ij}c^{links}_l]\!]$ participants obtain the embedding cost at the current RL iteration, in secret shared form. This cost is then summed up with $\sum_i [\![w^i_u]\!] + \sum_u [\![v_u]\!] + \sum_l [\![v_l]\!]$, which are penalty values corresponding to the feasibility of the solution. We remind that $v_u = 0$ (resp. $v_l = 0$) if node u's (resp., link l's) capacity is not exceeded (and ∞ otherwise) and $w^i_u = 1$ if node u is eligible to host VN_i (and ∞ otherwise).

The obtained value is then securely compared with the previous minimum cost (which is supposed to be initialized at $[\![\infty]\!]$) by means of the GE operator, whose output is recovered by the participants. If the current cost is less then the previous minimum (i.e., GE outputs 0), this cost is taken as the new minimum. The execution of the RL stops when the GE outputs 1 (i.e., current cost greater or equal to the previous minimum one) for a consecutive number of epochs equal to *patience*. Notice that the last iteration in which GE outputs 0 corresponds to the computation of the best embedding and the number of this iteration is known to all the participants.

7.7.4 Recovery of the Final Secrets

At the end of the execution of RL algorithm, the K ISPs deliver to the customer their shares of the minimum cost, obtained as described in the previous subsection. In this way, only the customer can recover the final embedding cost. Then, participants identify the iteration in which the best embedding has been obtained and exchange with each other the following data relative to that iteration:

Data Received by the customer the customer receives from all the K ISPs the shares relative to the states vector $\mathcal{S}_{CUST}^{(i)}, \forall i$ and $\mathcal{S}_{CUST}^{(kk\prime,ij)}, \forall kk\prime, ij$. From these data, customer discovers the ISPs in charge of embedding the VNs and the peering links traversed by the VPs.

Data Received by the ISPs The generic ISP_k receives from all the other participants the k-th component of vector state $\mathcal{S}_{CUST}^{(i)}, \forall i$, from which it can recover the information if VN_i has been assigned to it, and the vector state $\mathcal{S}_{ISP_k}^{(i)}, \forall i$, from which it discovers the physical node in which it has to embed VN_i. Then, the customer delivers to ISP_k the information on the peering nodes belonging to its infrastructure that should be traversed by $VP_{ij}, \forall ij$. Finally, ISP_k receives from all the participants the vector state $\mathcal{S}_{ISP_k}^{(uv,ij)}, \forall uv, ij$. Knowing the physical nodes that host the VNs assigned to it and the peering nodes traversed by the VPs, ISP_k is able to identify the pair of nodes uv and, from $\mathcal{S}_{ISP_k}^{(uv,ij)}$, discover the links in which VP_{ij} has to be embedded.

7.7.5 Fulfilliment of Privacy Requirements

7.7.5.1 Customer's Privacy Requirements

All the ISPs perform operations on data relative to customer's environments $\mathcal{E}_{CUST}^{(i)}, \forall i$ and $\mathcal{E}_{CUST}^{kk\prime,ij}, \forall kk\prime, ij$. These operations, described in subsection 7.7.2, are based on secure primitives. Hence, no information about the VG is leaked beyond the number of VNs N. At every iteration of the RL algorithm the participants discover if the current embedding cost is greater or equal to the previous minimum one, as described in subsection 7.7.3, which does not provide additional information about the VG. Finally, during the secret recovery phase described in subsection 7.7.4, each ISP only receives data related to the portion of VG that it has to embed, from which it derives the computational capacity d_i and type δ_i only of the generic VN_i, for every VN_i assigned to it. Similarly, each ISP discovers the bandwidth requirement d_{ij} for every VP_{ij} that it has to embed, but no information about other VPs. Hence, customer's privacy requirements are fulfilled.

7.7.5.2 ISPs' Privacy Requirements

Each participant performs operations over the shares hiding infrastructural datails of the generic ISP_k and from environments $\mathcal{E}_{ISP_k}^{(i)}, \forall i$ and $\mathcal{E}_{ISP_k}^{(uv,ij)}, \forall uv, ij$. From them, it is possible to discover the

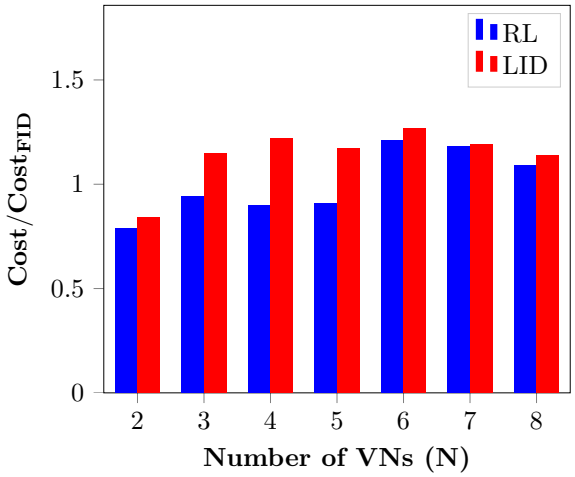


FIGURE 7.3: Comparison of costs achieved with the non-private RL, LID and FID approaches

number of nodes $|\mathcal{M}_k|$ and links $|\mathcal{L}_k|$ of ISP_k, as well as $|\mathbf{P}_k^{uv}|$, i.e., the number of paths connecting any two generic nodes u and v. However, since the operations performed on the shares are proved secure, no additional information about costs and capacities of ISP_k's nodes and links are exposed, as well as if there is or not a link between two generic nodes. Hence, also ISPs' privacy requirements are satisfied.

7.8 Numerical Results

7.8.1 Simulation Settings

In our experiments, we compare the RL-based approach with the LID and FID heuristics [41] considering the final embedding cost. We perform simulations based on the parameters presented in Table 7.3 and we show the results obtained by averaging the embedding costs of 50 experiments for each type of simulated scenario (e.g., characterized by a certain number of VNs N, number of ISPs K, etc...). Unless stated otherwise, we consider $K = 5$ ISPs with an average number of $M_k = 15$ nodes. The training of the RL algorithm is stopped when no improvements to the final costs are observed for a number of iteration $patience = 50000$.

7.8.2 Evaluation of the RL approach

In this subsection, we evaluate the effectiveness of the RL-based approach presented in Section 7.5 (i.e., the non privacy-preserving one). Experiments are performed setting $\mathcal{T}_{CUST}^{VN} = 20, \mathcal{T}_{CUST}^{VP} = 50, \mathcal{T}_{ISP}^{VN} = 1, \mathcal{T}_{ISP}^{VP} = 1$. In Fig. 7.3 we show $\frac{Cost_{RL}}{Cost_{FID}}$ and $\frac{Cost_{LID}}{Cost_{FID}}$ for $N \in \{2, ..., 8\}$. We observe that the $Cost_{RL}$ is always lower than $Cost_{LID}$ and, with $N < 6$, also then $Cost_{FID}$. Except for $N = 2$,

Table 7.3: Simulation Settings

Variable	Value
d_i	$\mathcal{U} \sim [0, 10]$
Probability that VN_i and VN_j exchange traffic	0.5
d_{ij}	$\mathcal{U} \sim [1, 10]$
Number of VNs Types	10
Probability that an ISP can host a certain type of VN	0.5
Probability that a physical node can host a certain type of VN	0.5
Number of Peering Nodes per ISP (on average))	1.5
Number of Outgoing Peering Links per ISP (on average)	4
Cost of embedding VN_i in node u	$\mathcal{U} \sim [1, 10]$
Computational capacity of node u	$\mathcal{U} \sim \{30, 40, 50\}$
Cost of embedding VP_{ij} in an internal link	$\mathcal{U} \sim \{6, 7, 8, 9\}$
Cost of embedding VP_{ij} in a peering link	$\mathcal{U} \sim \{11, 12, 13, 14, 15\}$
Capacity of an internal link	$\mathcal{U} \sim \{50, 100\}$
Capacity of a peering link	$\mathcal{U} \sim \{200, 400\}$
Learning rate lr and Discount Factor γ	0.125

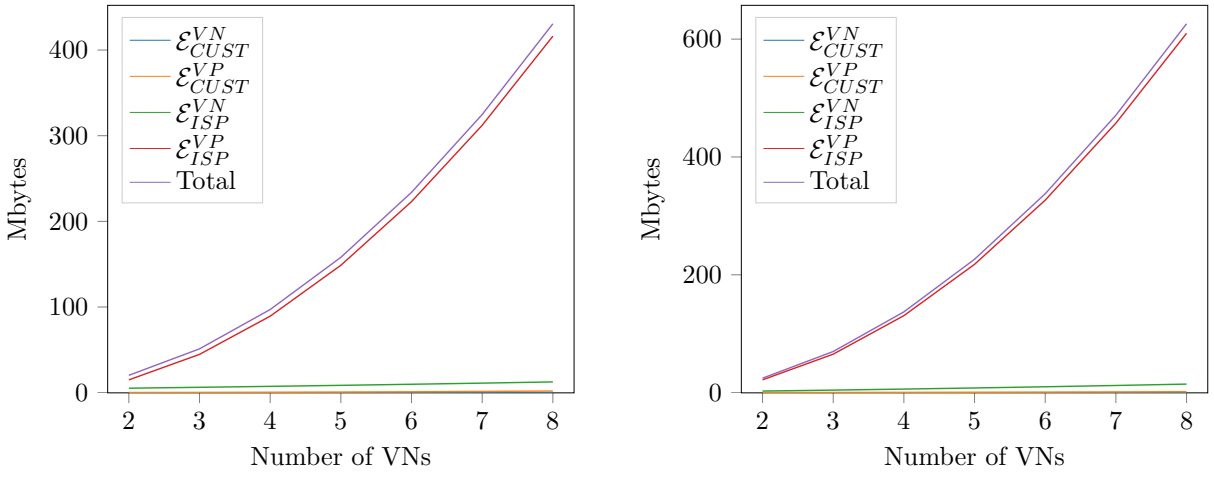
$\frac{Cost_{LID}}{Cost_{FID}}$ is always higher that 1 and, on average, $\frac{Cost_{LID}}{Cost_{FID}} = 1.14$ and $\frac{Cost_{RL}}{Cost_{FID}} = 1$, thus suggesting that the RL approach is a valid alternative to both the considered heuristics.

Then, Fig. 7.5a shows the minimum embedding cost as a function of the number of executed iterations of the RL algorithm for $N \in \{2, ..., 8\}$. As expected, a longer exploration of the solution space is needed to find the best embedding with increasing N. More specifically, the largest decrease of embedding cost is obtained after 25000 iterations for large instances of the VG (i.e., $N \geq 5$), while much fewer epochs are needed for small VGs (e.g., the minimum embedding cost of $N = 2$ is generally achieved after as few as 5000 epochs).

In the next subsection, we discuss the data overhead introduced by the privacy-preserving version of the RL algorithm considered in the aforementioned experiments.

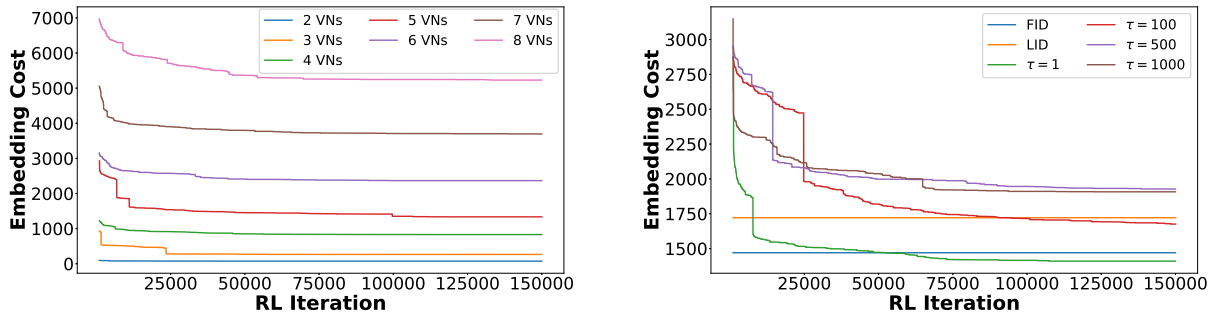
7.8.3 Data Overhead of the Privacy-Preserving RL

In Fig. 7.4 we show the volume of data that each pair of participants exchange with each other at every iteration of the privacy-preserving RL algorithm (where secret shares are assumed to be



(a) *Overhead of data exchanged on-line* (b) *Overhead of data exchanged off-line*

FIGURE 7.4: Cumulative Overhead in each type of Environment



(a) *Non-Privacy-Preserving RL* (b) *Privacy-Preserving RL ($N = 5$ VNs)*

FIGURE 7.5: Minimum Cost of the RL algorithm as a function of the number of iteration

represented using 20 bits). In particular, we show the overhead generated by performing operations in each one of the four considered types of environments (described in subsection 7.5.1) with increasing N, both on-line (in Fig. 7.4a) and off-line (in Fig. 7.4b).

First of all, we observe that the overhead increases with increasing N in all the environments, and this increase is much more voluminous in the environments related to the embedding of a VP into the physical links, i.e., $\mathcal{E}^{VP}_{ISP}$. As shown in Fig. 7.4a, the operations performed in this environment and described in Subsection 7.7.2.2 introduce, at every iteration of the RL algorithm, a cumulative on-line overhead of up $400 Mbytes$. On the other hand, operations in the other environments are much less expensive (e.g., operations in $\mathcal{E}^{VN}_{CUST}$ introduce $\sim 10^{-3} Mbytes$ per iteration). A similar trend can be observed for data exchanged off-line. Notice that these high values are mainly due to the fact that, as data is ciphered, participants are not aware of the specific environment in which they

have to act and operations are repeated in all environments. However, we remind that operations are effective only in the environment associated with a $[\![mask]\!] = [\![1]\!]$, as explained in Section 7.7.1. A strategy to reduce this high overhead is to limit the number of operations performed in environments of type $\mathcal{E}_{ISP}^{VP}$, as discussed in the next subsection.

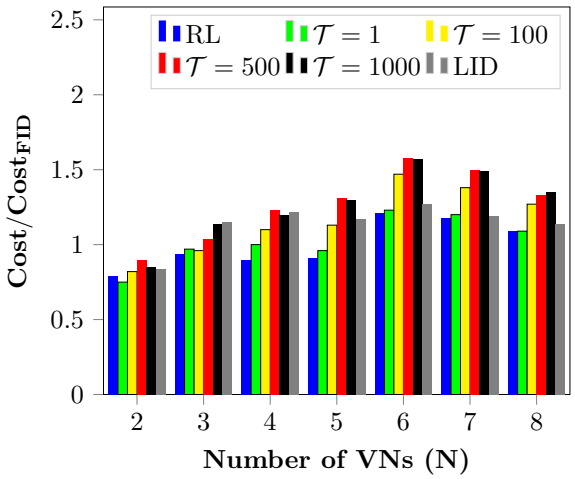
7.8.4 Comparison between Privacy-Preserving RL and the baselines

Here we evaluate the embedding cost achieved by performing operations on $\mathcal{E}_{ISP_k}^{uv,ij}, \forall k, uv, ij$ every $\mathcal{T}_{ISP}^{VP}$ iterations of the privacy-preserving RL algorithm, with $\mathcal{T}_{ISP}^{VP} \in \{1, 100, 500, 1000\}$, i.e., by varying the frequency of operations within the type of environments responsible for the greatest portion of the overhead. Obtained results are shown in Fig. 7.6a.

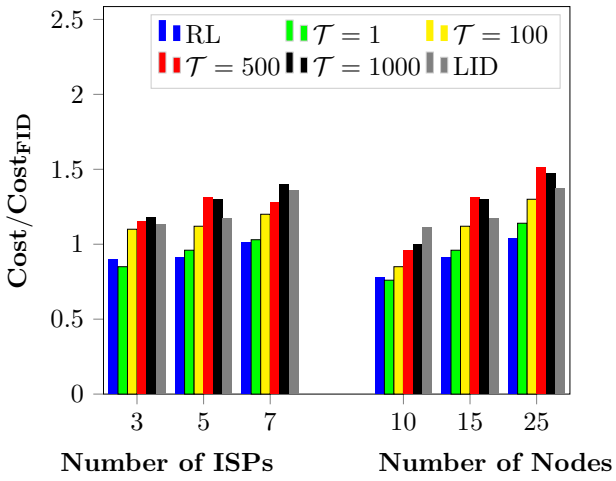
Firstly, we observe that the privacy-preserving RL yields to an average increase of the embedding cost of 3% with respect to FID heuristic, while the average costs of the non-privacy-preserving RL and FID are the same, when $\mathcal{T}_{ISP}^{VP} = 1$. This result can be explained considering that operations on ciphered data introduce several approximations (e.g., multiplications by decimal numbers are truncated to integer values, as mentioned in subsection 7.6.2.3). Then, we notice that the cost generally increases with increasing $\mathcal{T}_{ISP}^{VP}$, as the number of operations performed to find the physical paths that embed the virtual ones are reduced. In particular, there is an average increase of cost with respect to FID of 16%, 27% and 27%, achieved for $\mathcal{T}_{ISP}^{VP} = 100$,500 and 1000 iterations. For $N < 6$, the cost obtained with $\mathcal{T}_{ISP}^{VP} = 100$ is still lower than $Cost_{LID}$, which implies an advantage over to the privacy-preserving baseline (i.e., LID) at a significant reduction of data overhead. As an example, for $N = 5$, $\frac{Cost_{RL}}{Cost_{LID}}$ goes from 96% to 113% when $\mathcal{T}_{ISP}^{VP}$ goes from 1 to 100, which is an acceptable increase as the overhead at every iteration of the RL drops from 152.42 to 1.61 Mbytes (exchanged on-line) and from 223.12 to 2.31 Mbytes (exchanged off-line). In general, the total overhead decreases of a factor $\sim \mathcal{T}_{ISP}^{VP}$. On the other hand, for $N \geq 6$ the reduction of overhead achieved with $\mathcal{T}_{ISP}^{VP} = 100$ leads to an embedding cost that is higher than the LID heuristic. As a future study, we will evaluate such trade-off when $1 < \mathcal{T}_{ISP}^{VP} < 100$, which seems to be a crucial range to evaluate the ability of the privacy-preserving RL to effectively embed large VGs.

We then show in Fig. 7.5b the comparison of the minimum embedding cost as a function of the RL iteration, for several T_{ISP}^{VP}, considering $N = 5$. We observe that increasing T_{ISP}^{VP} does not significantly affect the number of iterations needed by the algorithm to converge but, as expected, reduces its ability to minimize the embedding cost.

Finally, we show in Fig. 7.6b the embedding cost considering $N = 5$ for several number of ISPs $K \in \{3, 5, 7\}$ (and fixed number of physical nodes $M_k = 15$) and for several number of average physical nodes in every ISP, i.e., $M_k \in \{10, 15, 25\}$ (and fixed number of ISPs $K = 5$). These results show that, when $T_{ISP}^{VP} \in \{1, 100\}$, the RL yields lower costs than the LID baseline if the number of ISPs (resp., of physical nodes) is increased from 5 to 7 (resp., from 10 to 25), suggesting the validity of the proposed RL approach in a broad range of scenarios.



(a) Comparison for several values of N (number of ISPs $K = 5$ and average number of internal nodes $|\mathcal{M}_k| = 15$)



(b) Comparison for several number of ISPs and number of nodes (number of VNs $N = 5$)

FIGURE 7.6: Comparison between LID, FID and privacy-preserving RL for several values of $\mathcal{T}_{ISP}^{VP}$

7.9 Conclusions

In this Chapter, we have proposed a privacy-preserving RL algorithm to perform VNE over a multi-domain infrastructure composed of several independent and mutually-distrustful ISPs. In this context, ISPs are not willing to expose details of their infrastructure that are needed to perform effective embedding (e.g., cost of traversing a link). By performing operations on secrets that ISPs and customer hide under the SSS scheme, our algorithm allows both customer and ISPs to retain

total privacy. We performed extensive simulations to evaluate the embedding cost achieved with our algorithm compared to two existing heuristics, i.e., the Limited Information Disclosure (LID) and the Full Information Disclosure (FID). By using the RL algorithm, we generally achieve embedding costs lower than both the baselines at the cost of a high data overhead exchanged between customer and ISPs. We then reduced the number of expensive operations and evaluated the resulting trade-off between overhead and embedding costs, showing that a considerable reduction of data overhead can be obtained while slightly increasing the final cost.

Conclusion 8

In this thesis we applied existing and novel privacy-preserving strategies in several Internet-based services. In particular, we considered the service of video content delivery jointly performed by CPs and ISPs, in which we identified various information that must be protected, such as the popularity of the contents offered by CPs, the location of users retrieving the contents and ISPs' infrastructural details, e.g., size of their cache servers. The proposed privacy-preserving strategies were mainly based on data perturbation, secure multi-party computation and secret sharing techniques. Considered use cases ranged from NN-compliant caching (for the realization of which we proposed an open and privacy-preserving protocol), privacy-preserving caching and privacy-preserving deployment of Virtual Servers based on users' location. Moreover, we proposed an awareness tool to give users the ability to measure and control the risks associated with the public exposition of their location in Online Social Networks. Finally, we proposed a Reinforcement Learning algorithm working on the SSS scheme to perform Virtual Network Embedding (VNE) over a multi-domain infrastructure in a privacy-preserving manner. In what follows we summarize more in detail the contribution of the thesis highlighting on the research issues considered per chapter.

- In Chapter 2 we investigated the privacy issues raised in the cooperation between a CP and an ISP that are willing to perform an efficient caching while keeping CP's video contents in encrypted form. To this end, we considered an existing architectural solution that allows the CP to hide its contents behind pseudonyms that the ISP is entitled to read to obtain information about contents' popularity (needed to perform caching efficiently) without decrypting the contents themselves. We formalized CP's privacy requirements and we observed that, by analyzing the occurrences of the pseudonyms, the ISP can still apply attacks that threaten them. We then formalized a pseudonyms' replacement strategy that the CP can apply to increase its privacy, and we noticed the existence of a trade-off between privacy and caching

effectiveness. Finally, we performed extensive simulations over both real and synthetic data to measure caching hit-ratio, average contents' retrieval latency and average traffic load on ISP's network links. Results suggest that privacy can be guaranteed at the cost of an acceptable degradation of caching effectiveness.

- In Chapter 3 we have observed that, given the large use of encryption applied by CPs (e.g., to protect their users' confidentiality), caching strategies need to be performed in cooperation by CPs and ISPs (as the latter, by themselves, cannot infer CPs' contents' popularity needed for effective caching). We considered several cooperative schemes and, by means of simulations, we shown that the employed cooperative approach may lead to a privileged treatment of a CP's traffic at the expenses of its competitors. Based on that, we advocated the inclusion of caching in the discussion on Network Neutrality and we proposed a possible definition of NN-compliant caching. In particular, our idea is that CPs should be allocated portions of caches' storage proportional to the popularity of their contents. As this definition threatens CPs' privacy (as they are not willing to reveal such business-critical information), we also advocated the design of a privacy-preserving protocol through which our vision of NN-compliant caching can be realized.

- In Chapter 4 we proposed a privacy-preserving protocol to implement NN-compliant caching, according to the definition provided in the previous Chapter. More specifically, this protocol enables an ISP to subdivide its caches' storage among several CPs proportionally to the popularity of their contents, while not requiring the exposition of sensitive information, namely the popularity of CPs' contents and the sizes of ISP's caches. The protocol achieves this objective by performing operations on data that CPs and ISP encrypt by means of the Shamir Secret Sharing Scheme. We validated the popularity-driven caches' subdivision against two baselines, i.e., the static subdivision (in which all the CPs are allocated the same portion of storage, regardless of their contents' popularity) and the Resource-Occupation (RO) driven subdivision (in which CPs receive a portion of storage proportional to the RO generated in the network of the ISP). Results, obtained by means of extensive simulations, shown that the subdivision enabled by our protocol leads to higher caching performance with respect to both the baselines, measured considering both the Hit-Rate and the RO. Moreover, the data overhead generated by the use of the protocol was much less significant than the reduction of RO measured by the ISP. Finally, the protocol proved scalable with increasing number of CPs and volume of their contents.

- In Chapter 5 we focused on the effective and privacy-preserving deployment, within the ISP's network, of Virtual Servers (VSs) that a CP employs to deliver Live Videos (LVs) to its users. Effectiveness was measured considering the average contents' retrieval latency, while privacy was relative to the protection of users' location and requests. In particular, we noticed that

ISP and CP have complementary information about their users. Specifically, ISP precisely knows users' location (needed to perform traffic delivery) but do not know the content of users' requests (which are encrypted by the CP for security reasons). On the other hand, the CP can only estimate users' location but knows exactly the content of their requests. As the geographical distribution of LVs proved more localized with respect to traditional video content (e.g., LVs may become viral within small areas), this information is required to perform the optimal deployment of the VSs. We defined CPs' and ISPs' privacy requirements, and we applied an existing secure multi-party computation protocol to make the ISP only learn an aggregated information on the number of requests for a given contents coming from a specific area. We shown that this information is sufficient to perform an effective deployment but not to violate several primary ISPs' and CPs' privacy requirements. However, to guarantee more demanding privacy requirements, an additional data perturbation approach was needed. We evaluated the trade-off between level of data perturbation and effectiveness of the deployment, and we concluded that challenging privacy requirements can only be met at the cost of a severe increase of average retrieval latency.

- In Chapter 6 we proposed a privacy awareness tool that Twitter users can employ to measure and control the risk associated with the public exposition of their location. We defined the privacy of a user as the geographical distance between her actual location (i.e., expressed as a geo-tag attached to a published content) and the estimated location that an attacker may infer by analyzing publicly-available geo-tags. We simulated such attack by means of a novel deep learning architecture that we proposed. Obtained results suggested high privacy risks for the majority of the users, as 60% of them could be localized with an average error below 1 Km. To address this issue, we proposed privacy-preserving strategies based on data perturbation techniques, and we shown that they allowed to significantly increase users' privacy. We then trained a Random Forest algorithm to learn a model of this value of privacy as a combination of several users' features, such as her mobility, the frequency of her geo-tags and the level of applied data perturbation. This model can be used as a privacy awareness tool, as it provides accurate estimates of users' privacy and allows to measure the impact that each feature has on it. In line with the idea of increasing users' awareness concerning the privacy of their location on Twitter, we then considered, as study case, the trade-off between effectiveness of a proximity marketing Location Based Service and users' privacy, resulting from the application of the proposed data perturbation strategy. The main take-home message was that high level of users' privacy can be achieved while not significantly sacrificing the effectiveness of this service.

- In Chapter 7, we proposed a Reinforcement Learning algorithm to perform Virtual Network Embedding (VNE) over a multi-ISP infrastructure in a privacy-preserving manner. The considered scenario involved a customer willing to offer a virtual service to its users exploiting a multi-domain physical infrastructure owned by several independent ISPs. More specifically, the

objective of the customer was to minimize the deployment cost of a virtual graph representing the offered service over such multi-domain network. In this multi-domain context, the ISPs do not expose to the customer complete information about their networks, which are considered privacy-sensitive and business critical assets, but rather a limited and abstracted view of their infrastructures. This Limited Information Disclosure (LID) approach leads to a suboptimal embedding with respect to its Full Information Disclosure (FID) counterpart. The proposed algorithm achieved an embedding cost comparable with an existing FID heuristic, while guaranteeing total privacy to both the customer and the ISPs. In fact, the proposed algorithm was designed to perform meaningful operations on data that have been ciphered using the SSS scheme. The main drawback of the proposed algorithm was the high data overhead introduced. We addressed the problem by reducing the number of expensive operations that it performed. We then observed that such reduction led to an increase of the embedding cost. We evaluated this trade-off for several levels of this reduction and noticed that intermediate costs between the FID and LID approaches could be achieved while lowering the overhead of at least two orders of magnitude.

In conclusion, this work advanced the existing research on privacy-preserving cooperation in Internet. Strategies to enable an effective yet privacy-preserving cooperation between large stakeholders in Internet are expected to have growing relevance in the service delivery of tomorrow. Given the importance and vastness of the considered topic, the approaches that we proposed should not be seen as the definitive solution to the privacy/utility dilemma, but rather as simple starting points. In particular, we advocate any contribution that may lead to their improvement! Specifically, several drawbacks may hinder the actual implementation of our solutions in a real scenario. In the following, we describe the drawbacks that we identified, along with possible research directions to get rid of them.

- Being privacy a rather problem-dependent concept, we could not identify a universal metric to measure its violation. In general, the provided definitions/metrics of privacy are meant to measure the reduction of information leakage given by the application of the proposed privacy-preserving strategies. As such measures strongly depend on the ability of the attacker to extract valuable information from an information source, we advocate the development of more advanced attack strategy to further validate the effectiveness of the proposed privacy-preserving protocols.

- Several of the proposed privacy-preserving approaches (namely, the protocol used to compute a NN-compliant caches' subdivision and the RL algorithm, described in Chapters 4 and 7, respectively) introduce non-negligible overhead in computational time and volume of data exchanged among the participants. A possible research direction to solve these issues is the

definition of similar protocols/algorithms based on lighter secret sharing strategies (e.g., trivial secret sharing).

Bibliography

[1] Bengt Ahlgren et al. A survey of information-centric networking. *IEEE Communications Magazine*, 50(7):26–36, 2012.

[2] Rodolfo Alvizu, Sebastian Troia, Guido Maier, and Achille Pattavina. Machine-learning-based prediction and optimization of mobile metro-core networks. In *2018 IEEE Photonics Society Summer Topical Meeting Series (SUM)*, pages 155–156. IEEE, 2018.

[3] Davide Andreoletti, Omran Ayoub, Silvia Giordano, Giacomo Verticale, and Massimo Tornatore. Privacy-preserving caching in isp networks. In *2019 IEEE 20th International Conference on High Performance Switching and Routing (HPSR)*, pages 1–6. IEEE, 2019.

[4] Davide Andreoletti et al. Discovering the geographic distribution of live videos' users: a privacy-preserving approach. In *2018 Global Communications Conference*, pages 1–6, 2018.

[5] Davide Andreoletti et al. An open privacy-preserving and scalable protocol for a network-neutrality compliant caching. In *Accepted for publication in proceedings of the IEEE International Conference on Communications (ICC 2019), 20-24 May, 2019, Shanghai, China.*, 2019.

[6] Davide Andreoletti, Cristina Rottondi, Silvia Giordano, Giacomo Verticale, and Massimo Tornatore. An open privacy-preserving and scalable protocol for a network-neutrality compliant caching. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[7] Davide aand others Andreoletti. To be neutral or not neutral? the in-network caching dilemma. *IEEE Internet Computing*, 2018.

[8] Ji Ao, Peng Zhang, and Yanan Cao. Estimating the locations of emergency events from twitter streams. *Procedia Computer Science*, 31:731–739, 2014.

[9] Andrea Araldo, Gyorgy Dan, and Dario Rossi. Caching encrypted content via stochastic cache partitioning. *IEEE/ACM Transactions on Networking (TON)*, 26(1):548–561, 2018.

[10] Andrea Araldo, Gyorgy Dan, and Dario Rossi. Caching encrypted content via stochastic cache partitioning. *IEEE/ACM Transactions on Networking (TON)*, 26(1):548–561, 2018.

[11] Samuel MA Araújo, Fernanda SH de Souza, and Geraldo R Mateus. Virtual network embedding in multi-domain environments with energy efficiency concepts. In *2018 International Conference on Information Networking (ICOIN)*, pages 205–210. IEEE, 2018.

[12] Omran Ayoub, Francesco Musumeci, Davide Andreoletti, Marco Mussini, Massimo Tornatore, and Achille Pattavina. Optimal cache deployment for video-an-demand delivery in optical metro-area networks. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2018.

[13] Omran Ayoub, Francesco Musumeci, Massimo Tornatore, and Achille Pattavina. Efficient routing and bandwidth assignment for inter-data-center live virtual-machine migrations. *IEEE/OSA Journal of Optical Communications and Networking*, 9(3):B12–B21, 2017.

[14] Omran Ayoub, Francesco Musumeci, Massimo Tornatore, and Achille Pattavina. Techno-economic evaluation of cdn deployments in metropolitan area networks. In *2017 International Conference on Networking and Network Applications (NaNA)*, pages 314–319. IEEE, 2017.

[15] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th international conference on World Wide Web*, pages 181–190. ACM, 2007.

[16] James P Bagrow, Xipei Liu, and Lewis Mitchell. Information flow reveals prediction limits in online social activity. *Nature Human Behaviour*, 3(2):122, 2019.

[17] Marco Balduzzi, Christian Platzer, Thorsten Holz, Engin Kirda, Davide Balzarotti, and Christopher Kruegel. Abusing social networks for automated user profiling. In *International Workshop on Recent Advances in Intrusion Detection*, pages 422–441. Springer, 2010.

[18] Benjamin Baron and Mirco Musolesi. Interpretable machine learning for privacy-preserving pervasive systems. *arXiv preprint arXiv:1710.08464*, 2017.

[19] Donald Beaver. Efficient multiparty protocols using circuit randomization. In *Annual International Cryptology Conference*, pages 420–432. Springer, 1991.

[20] Donald Beaver. Efficient multiparty protocols using circuit randomization. In *Annual International Cryptology Conference*, pages 420–432. Springer, 1991.

[21] Alastair R Beresford, Andrew Rice, Nicholas Skehin, and Ripduman Sohan. Mockdroid: trading privacy for application functionality on smartphones. In *Proceedings of the 12th workshop on mobile computing systems and applications*, pages 49–54. ACM, 2011.

[22] Alastair R Beresford and Frank Stajano. Mix zones: User privacy in location-aware services. In *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*, pages 127–131. IEEE, 2004.

[23] Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Optimal geo-indistinguishable mechanisms for location privacy. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 251–262. ACM, 2014.

[24] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web caching and zipf-like distributions: Evidence and implications. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 126–134. IEEE, 1999.

[25] Lee Breslau et al. Web caching and zipf-like distributions: Evidence and implications. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 126–134. IEEE, 1999.

[26] Haotong Cao, Han Hu, Zhicheng Qu, and Longxiang Yang. Heuristic solutions of virtual network embedding: A survey. *China Communications*, 15(3):186–219, 2018.

[27] Octavian Catrina and Sebastiaan De Hoogh. Improved primitives for secure multiparty integer computation. In *International Conference on Security and Cryptography for Networks*, pages 182–199. Springer, 2010.

[28] Abdelberi Chaabane, Gergely Acs, Mohamed Ali Kaafar, et al. You are what you like! information leakage through users' interests. In *Proceedings of the 19th Annual Network & Distributed System Security Symposium (NDSS)*. Citeseer, 2012.

[29] Wei Koong Chai et al. Cache "less for more" in information-centric networks. In *International Conference on Research in Networking*, pages 27–40. Springer, 2012.

[30] Qingjun Chen, Shouqian Shi, Xin Li, Chen Qian, and Sheng Zhong. Sdn-based privacy preserving cross domain routing. *IEEE Transactions on Dependable and Secure Computing*, 2018.

[31] Qingjun Chen, Shouqian Shi, Xin Li, Chen Qian, and Sheng Zhong. Sdn-based privacy preserving cross domain routing. *IEEE Transactions on Dependable and Secure Computing*, 2018.

[32] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 759–768. ACM, 2010.

[33] Wen-Haw Chong and Ee-Peng Lim. Exploiting contextual information for fine-grained tweet geolocation. In *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[34] Cisco. Cisco visual networking index: Forecast and trends, 2017–2022. *White Paper*, 1, 2018.

[35] Shujie Cui et al. Multi-cdn: Towards privacy in content delivery networks. *IEEE Transactions on Dependable and Secure Computing*, 2018.

[36] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. *Journal of Computer Security*, 19(5):895–934, 2011.

[37] Ivan Damgård, Valerio Pastro, Nigel Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *Annual Cryptology Conference*, pages 643–662. Springer, 2012.

[38] Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. Fast and private computation of cardinality of set intersection and union. In *CANS*, pages 218–231. Springer, 2012.

[39] Tim Dierks and Eric Rescorla. The transport layer security (tls) protocol version 1.2. Technical report, 2008.

[40] David Dietrich, Amr Rizk, and Panagiotis Papadimitriou. Multi-domain virtual network embedding with limited information disclosure. In *2013 IFIP Networking Conference*, pages 1–9. IEEE, 2013.

[41] David Dietrich, Amr Rizk, and Panagiotis Papadimitriou. Multi-provider virtual network embedding with limited information disclosure. *IEEE Transactions on Network and Service Management*, 12(2):188–201, 2015.

[42] Suguo Du, Xiaolong Li, Jinli Zhong, Lu Zhou, Minhui Xue, Haojin Zhu, and Limin Sun. Modeling privacy leakage risks in large-scale social networks. *IEEE Access*, 6:17653–17665, 2018.

[43] Xavier Dutreilh, Sergey Kirgizov, Olga Melekhova, Jacques Malenfant, Nicolas Rivierre, and Isis Truck. Using reinforcement learning for autonomic resource allocation in clouds: towards a fully automated workflow. In *ICAS 2011, The Seventh International Conference on Autonomic and Autonomous Systems*, pages 67–74, 2011.

[44] Anne Edmundson et al. Ocdn: oblivious content distribution networks. *arXiv preprint arXiv:1711.01478*, 2017.

[45] Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1277–1287. Association for Computational Linguistics, 2010.

[46] Salah-Eddine Elayoubi and James Roberts. Performance and cost effectiveness of caching in mobile access networks. In *Proceedings of the 2nd International Conference on Information-Centric Networking*, pages 79–88. ACM, 2015.

[47] Yehia Elkhatib, , et al. Dataset on usage of a live & vod p2p iptv service. In *Peer-to-Peer Computing (P2P), 14-th IEEE International Conference on*, pages 1–5. IEEE, 2014.

[48] Jeffrey Erman, Alexandre Gerber, KK Ramadrishnan, Subhabrata Sen, and Oliver Spatscheck. Over the top video: the gorilla in cellular networks. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 127–136. ACM, 2011.

[49] Andreas Fischer, Juan Felipe Botero, Michael Till Beck, Hermann De Meer, and Xavier Hesselbach. Virtual network embedding: A survey. *IEEE Communications Surveys & Tutorials*, 15(4):1888–1906, 2013.

[50] Anna Förster, Kamini Garg, Hoang Anh Nguyen, and Silvia Giordano. On context awareness and social distance in human mobility traces. In *Proceedings of the third ACM international workshop on Mobile Opportunistic Networks*, pages 5–12. ACM, 2012.

[51] Benjamin Frank et al. *Content-aware traffic engineering*, volume 40. ACM, 2012.

[52] Benjamin Frank, Ingmar Poese, Yin Lin, Georgios Smaragdakis, Anja Feldmann, Bruce Maggs, Jannis Rake, Steve Uhlig, and Rick Weber. Pushing cdn-isp collaboration to the limit. *ACM SIGCOMM Computer Communication Review*, 43(3):34–44, 2013.

[53] David Garcia. Privacy beyond the individual. *Nature Human Behaviour*, 3(2):112, 2019.

[54] Thiago Garrett, Ligia E Setenareski, Leticia M Peres, Luis CE Bona, and Elias P Duarte. Monitoring network neutrality: A survey on traffic differentiation detection. *IEEE Communications Surveys & Tutorials*, 20(3):2486 – 2517, 2018.

[55] Mark Graham, Scott A Hale, and Devin Gaffney. Where in the world are you? geolocation and language identification in twitter. *The Professional Geographer*, 66(4):568–578, 2014.

[56] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM, 2003.

[57] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):93, 2018.

[58] Kailing Guo, Ying Wang, Xuesong Qiu, Wenjing Li, and Ailing Xiao. Particle swarm optimization based multi-domain virtual network embedding. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 798–801. IEEE, 2015.

[59] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[60] Nicolas Herbaut, Daniel Négru, Yiping Chen, Pantelis A Frangoudis, and Adlen Ksentini. Content delivery networks as a virtual network function: a win-win isp-cdn collaboration. In *IEEE Global Communications Conference (GLOBECOM), 2016*, pages 1–6, 2016.

[61] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[62] Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 161–171. ACM, 2007.

[63] Ines Houidi, Wajdi Louati, Walid Ben Ameur, and Djamal Zeghlache. Virtual network provisioning across multiple substrate networks. *Computer Networks*, 55(4):1011–1023, 2011.

[64] Qingmin Jia, Renchao Xie, Tao Huang, Jiang Liu, and Yunjie Liu. The collaboration for content delivery and network infrastructures: A survey. *IEEE Access*, 5:18088–18106, 2017.

[65] Garrett Johnson, Randall A Lewis, and David Reiley. Location, location, location: repetition and proximity increase advertising effectiveness. *Available at SSRN 2268215*, 2016.

[66] Kristján Valur Jónsson, Gunnar Kreitz, and Misbah Uddin. Secure multi-party sorting and applications. *IACR Cryptology ePrint Archive*, 2011:122, 2011.

[67] Mostafa Karimzadeh, Zhongliang Zhao, Florian Gerber, and Torsten Braun. Mobile users location prediction with complex behavior understanding. In *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, pages 323–326. IEEE, 2018.

[68] Ruimin Ke, Wan Li, Zhiyong Cui, and Yinhai Wang. Two-stream multi-channel convolutional neural network (tm-cnn) for multi-lane traffic speed prediction considering traffic volume impact. *arXiv preprint arXiv:1903.01678*, 2019.

[69] Sheila Kinsella, Vanessa Murdock, and Neil O'Hare. I'm eating a sandwich in glasgow: modeling locations with tweets. In *Proceedings of the 3rd international workshop on Search and mining user-generated contents*, pages 61–68. ACM, 2011.

[70] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

[71] Jan Krämer, Lukas Wiewiorra, and Christof Weinhardt. Net neutrality: A progress report. *Telecommunications Policy*, 37(9):794–813, 2013.

[72] Nikolaos Laoutaris et al. The lcd interconnection of lru caches and its analysis. *Performance Evaluation*, 63(7):609–634, 2006.

[73] P Laud and L Kamm. Practical applications of secure multiparty computation. *Applications of Secure Multiparty Computation*, 13:246, 2015.

[74] Jérémie Leguay, , et al. Cryptocache: Network caching with confidentiality, 2017.

[75] Kai Li and Timon C Du. Building a targeted mobile advertising system for location-based services. *Decision Support Systems*, 54(1):1–8, 2012.

[76] Xin Li, Zhuzhong Qian, Sanglu Lu, and Jie Wu. Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center. *Mathematical and Computer Modelling*, 58(5-6):1222–1235, 2013.

[77] Yan Li, Yingjiu Li, Qiang Yan, and Robert H Deng. Privacy leakage analysis in online social networks. *Computers & Security*, 49:239–254, 2015.

[78] John Lingad, Sarvnaz Karimi, and Jie Yin. Location extraction from disaster-related microblogs. In *Proceedings of the 22nd international conference on world wide web*, pages 1017–1020. ACM, 2013.

[79] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[80] Jiayi Liu, Qinghai Yang, Gwendal Simon, and Weili Cui. Migration-based dynamic and practical virtual streaming agent placement for mobile adaptive live streaming. *IEEE Transactions on Network and Service Management*, 2017.

[81] Ximeng Liu, Robert Deng, Kim-Kwang Raymond Choo, and Yang Yang. Privacy-preserving reinforcement learning design for patient-centric dynamic treatment regimes. *IEEE Transactions on Emerging Topics in Computing*, 2019.

[82] Luca Luceri, Torsten Braun, and Silvia Giordano. Analyzing and inferring human real-life behavior through online social networks with social influence deep learning. *Applied Network Science*, 4(1):34, 2019.

[83] Xueming Luo, Michelle Andrews, Zheng Fang, and Chee Wei Phang. Mobile targeting. *Management Science*, 60(7):1738–1756, 2013.

[84] Patrick Maillé, Gwendal Simon, and Bruno Tuffin. Toward a net neutrality debate that conforms to the 2010s. *IEEE Communications Magazine*, 54(3):94–99, 2016.

[85] Michele Mangili et al. Performance analysis of content-centric and content-delivery networks with evolving object popularity. *Computer Networks*, 94:80–98, 2016.

[86] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[87] Marcio Melo, Susana Sargento, Ulrich Killat, Andreas Timm-Giel, and Jorge Carapinha. Optimal virtual network embedding: Node-link formulation. *IEEE Transactions on Network and Service Management*, 10(4):356–368, 2013.

[88] Dominik Molitor, Philipp Reichhart, Martin Spann, and Anindya Ghose. Measuring the effectiveness of location-based advertising: A randomized field experiment. *Available at SSRN 2645281*, 2019.

[89] Steven Mudda, Matteo Zignani, Sabrina Gaito, Silvia Giordano, and Gian Paolo Rossi. Timely and personalized services using mobile cellular data. *Online Social Networks and Media*, 2019.

[90] Cisco Visual Networking. Cisco global cloud index: Forecast and methodology 2015–2020. *White paper*, 2016.

[91] Cisco Visual networking Index. Forecast and methodology, 2016-2021, white paper. *San Jose, CA, USA*, 1, 2016.

[92] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. Mining user mobility features for next place prediction in location-based services. In *Data mining (ICDM), 2012 IEEE 12th international conference on*, pages 1038–1043. IEEE, 2012.

[93] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 1999.

[94] Michela Papandrea, Karim Keramat Jahromi, Matteo Zignani, Sabrina Gaito, Silvia Giordano, and Gian Paolo Rossi. On the properties of human mobility. *Computer Communications*, 87:19–36, 2016.

[95] David Pariag and Tim Brecht. Application bandwidth and flow rates from 3 trillion flows across 45 carrier networks. In *International Conference on Passive and Active Network Measurement*, pages 129–141. Springer, 2017.

[96] Beatrice Perez, Mirco Musolesi, and Gianluca Stringhini. You are your metadata: Identification and obfuscation of social media users using metadata information. In *Twelfth International AAAI Conference on Web and Social Media*, 2018.

[97] Ingmar Poese, Benjamin Frank, Georgios Smaragdakis, Steve Uhlig, Anja Feldmann, and Bruce Maggs. Enabling content-aware traffic engineering. *ACM SIGCOMM Computer Communication Review*, 42(5):21–28, 2012.

[98] Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. Ip geolocation databases: Unreliable? *ACM SIGCOMM Computer Communication Review*, 41(2):53–56, 2011.

[99] Iasonas Polakis, George Argyros, Theofilos Petsios, Suphannee Sivakorn, and Angelos D Keromytis. Where's wally?: Precise user discovery attacks in location proximity services. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 817–828. ACM, 2015.

[100] Aravindh Raman, Gareth Tyson, and Nishanth Sastry. Facebook (a) live?: Are live social broadcasts really broad casts? In *Proceedings of the 2018 world wide web conference*, pages 1491–1500. International World Wide Web Conferences Steering Committee, 2018.

[101] Garima Rastogi, Satya Narayan, Gopal Krishan, and Rama Sushil. Deployment of cloud using open-source virtualization: Study of vm migration methods and benefits. In *Big Data Analytics*, pages 553–563. Springer, 2018.

[102] Peter Rost, Christian Mannweiler, Diomidis S Michalopoulos, Cinzia Sartori, Vincenzo Sciancalepore, Nishanth Sastry, Oliver Holland, Shreya Tayade, Bin Han, Dario Bega, et al. Network slicing to enable scalability and flexibility in 5g mobile networks. *IEEE Communications magazine*, 55(5):72–79, 2017.

[103] Adam Sadilek, Henry Kautz, and Jeffrey P Bigham. Finding your friends and following them to where you are. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 723–732. ACM, 2012.

[104] Lorenzo Saino et al. Hash-routing schemes for information centric networking. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-Centric Networking (ICN '13)*, August 2013.

[105] Fady Samuel, Mosharaf Chowdhury, and Raouf Boutaba. Polyvine: policy-based virtual network embedding across multiple domains. *Journal of Internet Services and Applications*, 4(1):6, 2013.

[106] Lalitha Sankar, S Raj Rajagopalan, and H Vincent Poor. A theory of utility and privacy of data sources. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 2642–2646. IEEE, 2010.

[107] Axel Schulz, Aristotelis Hadjakos, Heiko Paulheim, Johannes Nachtwey, and Max Mühlhäuser. A multi-indicator approach for geolocalization of tweets. In *Seventh international AAAI conference on weblogs and social media*, 2013.

[108] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[109] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[110] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. Quantifying location privacy. In *2011 IEEE symposium on security and privacy*, pages 247–262. IEEE, 2011.

[111] Neil Spring et al. Measuring isp topologies with rocketfuel. In *ACM SIGCOMM Computer Communication Review*, volume 32, pages 133–145. ACM, 2002.

[112] Tiina Turban. A secure multi-party computation protocol suite inspired by shamir's secret sharing scheme. Master's thesis, Institutt for telematikk, 2014.

[113] Barbara Van Schewick. Network neutrality and quality of service: What a nondiscrimination rule should look like. *Stan. L. Rev.*, 67:1, 2015.

[114] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 605–613. ACM, 2013.

[115] Xingliang Yuan et al. Enabling secure and efficient video delivery through encrypted in-network caching. *IEEE Journal on Selected Areas in Communications*, 34(8):2077–2090, 2016.

[116] Xingliang Yuan, Xinyu Wang, Jinfan Wang, Yilei Chu, Cong Wang, Jianping Wang, Marie-Jose Montpetit, and Shucheng Liu. Enabling secure and efficient video delivery through encrypted in-network caching. *IEEE Journal on Selected Areas in Communications*, 34(8):2077–2090, 2016.

[117] Fida-E Zaheer, Jin Xiao, and Raouf Boutaba. Multi-provider service negotiation and contracting in network virtualization. In *2010 IEEE Network Operations and Management Symposium-NOMS 2010*, pages 471–478. IEEE, 2010.

[118] Rui Zhao, Chuan Yue, and Qi Han. Cross-site input inference attacks on mobile web users. In *International Conference on Security and Privacy in Communication Systems*, pages 629–643. Springer, 2017.

[119] Elena Zheleva and Lise Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th international conference on World wide web*, pages 531–540. ACM, 2009.

[120] Xin Zheng, Jialong Han, and Aixin Sun. A survey of location prediction on twitter. *IEEE Transactions on Knowledge and Data Engineering*, 2018.