symmetry

MDPI

Article

Mitigation of Privacy Threats due to Encrypted Traffic Analysis through a Policy-Based Framework and MUD Profiles

Gianmarco Baldini [1,*], José L. Hernandez-Ramos [1], Slawomir Nowak [2], Ricardo Neisse [1] and Mateusz Nowak [2]

[1] Joint Research Centre, European Commission, 1050 Ispra , Italy; jose-luis.hernandez-ramos@ec.europa.eu (J.L.H.-R.); ricardo.neisse@ec.europa.eu (R.N.)
[2] Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences (IITiS PAN), 44-100 Gliwice , Poland; snowak@iitis.pl (S.N.); mateusz@iitis.pl (M.N.)
* Correspondence: gianmarco.baldini@ec.europa.com or gianmarco.baldini@ec.europa.eu; Tel.: +39-0332-78-6618

Received: 23 August 2020; Accepted: 17 September 2020; Published: 22 September 2020

Abstract: It has been proven in research literature that the analysis of encrypted traffic with statistical analysis and machine learning can reveal the type of activities performed by a user accessing the network, thus leading to privacy risks. In particular, different types of traffic (e.g., skype, web access) can be identified by extracting time based features and using them in a classifier. Such privacy attacks are asymmetric because a limited amount of resources (e.g., machine learning algorithms) can extract information from encrypted traffic generated by cryptographic systems implemented with a significant amount of resources. To mitigate privacy risks, studies in research literature have proposed a number of techniques, but in most cases only a single technique is applied, which can lead to limited effectiveness. This paper proposes a mitigation approach for privacy risks related to the analysis of encrypted traffic which is based on the integration of three main components: (1) A machine learning component which proactively analyzes the encrypted traffic in the network to identify potential privacy threats and evaluate the effectiveness of various mitigation techniques (e.g., obfuscation), (2) a policy based component where policies are used to enforce privacy mitigation solutions in the network and (3) a network node profile component based on the Manufacturer Usage Description (MUD) standard to enable changes in the network nodes in the cases where the first two components are not effective in mitigating the privacy risks. This paper describes the different components and how they interact in a potential deployment scenario. The approach is evaluated on the public dataset ISCXVPN2016 and the results show that the privacy threat can be mitigated significantly by removing completely the identification of specific types of traffic or by decreasing the probability of their identification as in the case of VOIP by 50%, Chat by 40% and Browsing by 33%, thus reducing significantly the privacy risk.

Keywords: machine learning; encrypted traffic; policy based framework; privacy

1. Introduction

Nowadays, most of the traffic on the Internet is communicated by using well-known security protocols, such as Transport Layer Security (TLS) [1] or IP Security (IPSec) [2], which provide basic security properties, such as confidentiality and integrity. The use of these protocols can ensure that an eavesdropper will not be able to decrypt the traffic content transmitted by a user or modify it without detection. In recent years, internet traffic is increasingly based on encryption technology to secure the internet connections (e.g., using HTTPS). One example of such trend is provided in [3] where encrypted traffic for several Google products has increased steadily since January 2014. However, as demonstrated

in literature [4], the use of machine learning algorithms could help identifying user activities, which could represent a privacy risk. Indeed, such techniques can be used to recognize different types of traffic, such as web access or videostreaming. The identification of a certain type of traffic can be generally obtained both through the analysis of the payload of encrypted traffic (i.e., deep packet inspection [5]) and based on the analysis of features extracted from the encrypted traffic. While deep packet inspection allows the identification of traffic types with high accuracy, many authors [6,7] have shown that even the analysis of statistical features extracted from the traffic provides a high classification accuracy and therefore it generates significant privacy threats. Deep packet inspection requires a significant computational effort by a malicious entity (i.e., an attacker) because the amount of encrypted traffic can be quite large. However, the statistical features approach can be implemented efficiently even with limited resources and it can be easier to implement. For this reason, our work focuses on this second approach. As a consequence, such privacy attacks are asymmetric attacks because a limited amount of resources (e.g., machine learning algorithms) can extract information from encrypted traffic generated by cryptographic systems implemented with a significant amount of resources (e.g., key management systems, cryptomodules and so on).

With the increasing use of modern machine learning techniques, encrypted traffic analysis has attracted a significant interest in recent years. Indeed, a recent report from the European Union Agency for Cybersecurity (ENISA) [8] highlights that modern encrypted traffic analysis is able to weaken the confidentiality property and gain information on the type of traffic (as already mentioned above) but it can also identify which website a user is surfing or which files a user downloads and shares over an encrypted channel. In general, existing works in literature have shown that the recent improvement of encrypted traffic analysis (due also to sophisticated signal processing and machine learning techniques, such as deep learning) may generate serious privacy risks and they may be implemented with relatively simple tools and computing platforms. Furthermore, once a certain privacy threat has been identified, privacy mitigation techniques must be enabled and activated to mitigate such threat. Toward this end, the use of automated and efficient techniques is crucial to react against new vulnerabilities. Indeed, the identification of a certain privacy threat could require the automatic adaptation of a network component's behavior to filter the communication with specific services.

The problem this paper tries to solve is to provide the capabilities to a network manager to identify in an efficient way potential privacy threats in the network and mitigate such threats in an automatic and comprehensive way. As described in the related work Section 2, there are no comprehensive frameworks which link the detection capabilities with the automatic enforcement of actions to mitigate the privacy threat. One specific issue is that a privacy threat may be mitigated only by denying the traffic flow, which can be a drastic measure as it implies an absence of service. Then, there is a need for a framework, which provides to the network manager a wider set of tools to mitigate the privacy threat while limiting (when possible) the impact on the provision of network traffic services.

Our Contribution: To cope with the identification and mitigation of privacy threats in encrypted traffic analysis, this work proposes a multi-pronged approach based on three different sets of components:

- The first component is a machine learning classifier, which inspects the encrypted traffic to identify the types of traffic to anticipate a similar privacy threat, which can be exploited by a malicious entity. The analysis of the encrypted traffic has been implemented on the ISCXVPN2016 dataset, which is publicly available [9].
- The second component is a policy based framework whose policies are activated once the machine learning component has detected a privacy threat. As it will be described in the rest of this paper, the policy is activated when the machine learning classifier has detected the possibility of a privacy threat with a high probability. In particular, our approach is based on an adaptation of the Seckit framework [10] to mitigate the privacy threat either by denying the access to specific features extracted from the encrypted traffic, or by filtering the traffic on specific ports in the most critical scenarios. In this case, the policy based framework is based on rules and logic statements,

beyond the classical use for access control restrictions based on the well-known eXtensible Access Control Markup Language (XACML) [11].

- The third component is based on the concept of node profile where the nodes of a fixed network like a Software Defined Network (SDN) and the connected devices (e.g., computers or IoT devices) are represented and controlled by the network manager through specific profiles. The concept of the node profile is based on the recent Manufacturer Usage Description (MUD) standard [12], which makes the proposed approach relatively agnostic to the characteristics of the network and its nodes and provides a direct control to the network manager to deny or limit the traffic in a node in the exceptional cases when the policy based framework is not able to fully mitigate the privacy threat.

We acknowledge that each of the components described above is not novel on its own and it has been applied separately to SDN, IoT contexts and Information and Communication Technologies (ICT) infrastructures in general. The relatively novel aspect of this paper is in the combination of the three elements to provide the capability to network managers to detect privacy threats in their network and mitigate them in one comprehensive framework.

The structure of this paper is as follows: Section 2 describes the state of art in research literature in different areas including mitigation of privacy threats due to traffic analysis, application of policy based frameworks as countermeasures to security and privacy threats in fixed networks and application of machine learning to the analysis of encrypted traffic. This section does also provide some background information on entrypted traffic. Section 3 describes the overall methodology and materials used in the analysis of the proposed framework. Then, the overall architecture is presented in Section 4 together with a description of the three main components: The policy based framework, MUD and machine learning classifiers of encrypted traffic. Section 5 describes a potential implementation of the framework to mitigate a specific privacy threat related to the possibility to identify the traffic types of a user. The section describes the templates of the policies, the description of the MUD profiles and the results from the machine learning analysis of the encrypted traffic. Performance aspects are also discussed in this section. Finally, Section 6 concludes this paper.

2. Related Work and Background Information

The aim of this section is to describe the related work in the field of analysis of encrypted traffic and the application of the other components identified in the Introduction section. Background information on encrypted traffic is presented in the sub-section Encrypted traffic Section 2.1 below.

The combination of policy based frameworks with the analysis of encrypted traffic and MUD has not been reported in literature yet and similar works on the mitigation of privacy threats in Software Defined Networks (SDN) are limited.

Then we focus this section on the analysis of the related work in three main areas: (a) On the application of policy-based frameworks to mitigate cybersecurity and more specifically privacy threats in fixed networks in Section 2.2, (b) on the application of the MUD concept in cybersecurity in Section 2.3 and (c) the analysis of encrypted traffic in Section 2.4.

2.1. Background Information on Encryption Algorithms for Network Traffic

Encryption protocols are used to provide security and privacy to network communication. By far the most popular network traffic encryption control is the Transport Layer Security (TLS) [8], which has the objective to implement confidentiality, authentication and message integrity on the network connection. Confidentiality prevents the access to the contents of the message by unauthorized parties, authentication verifies the identity and authenticate the parties involved in the secure communication and message integrity ensures that messages are not modified during the communication or that such modifications are noted. Confidentiality is directly linked to the mitigation of privacy risks because the exchange content is not accessible to any parties, who do not have the encryption credentials.

Hypertext Transfer Protocol Secure (HTTPS) is an extension of the Hypertext Transfer Protocol (HTTP), which is used for secure communication over a computer network. In HTTPS, the communication protocol is encrypted using TLS. Then, the entirety of the HTTP protocol can be encrypted. One key issue is how web browsers (or other parties) know how to trust HTTPS websites. This is implemented using certificates trusted from Certification Authorities (CA) and public key cryptographic algorithms (e.g., Elliptic Curve Diffie-Hellman protocols). The certificates are defined by using the X.509 standard and they are provided by trusted CAs, which issue a certificate binding a public key to particular distinguished name and which are part of a Public Key Infrastructure (PKI).

2.2. Policy-Based Frameworks to Mitigate Security and Privacy Threats to Networks

Policy-based frameworks are mostly used in literature for access control and network management. In SDN, policy languages have been created to implement network management functions but not specifically to mitigate privacy threat. A review of the SDN policy languages is provided in [13].

In general (apart from the SDN context), one of the most common implementations of policy based frameworks is the eXtensible Access Control Markup Language (XACML), which is an OASIS standard [11] designed to replace application-specific and proprietary access control policy languages. A XACML policy can be defined to control the access to an entity (e.g., a network traffic application) which want to perform an action on a resource like a service of a dataset. When an entity would like to access a resource, the Policy Enforcement Point (PEP) regulate the access to the protected resource on a basis of a specific policy and a request/response protocol defined in the XACML language. The decision to identify a specific policy to allow or deny the access to the resource is taken by a Policy Decision Point (PDP), which stores or have access to a Policy repository, where the policies are stored. The decision taken by the PDP can be based on various logical or temporal conditions. XACML has been used in Internet of Things (IoT) context in [14]. Furthermore, the National Institute of Standards and Technology (NIST) defined the Next Generation Access Control (NGAC), which represents an alternative approach for the definition of access control policies by using a generic set of relations and functions. NGAC is described and compared with XACML in [15].

Beyond access control, policy-based frameworks like the SecKit [10] have been used in a more comprehensive way to express usage control rules through policies that regulate the behavior of system elements. Indeed, unlike traditional policy-based approaches for access control, Seckit provides a framework for the specification of different security properties in a certain system. Thus, the SecKit policy language can be used to define rules not only based on the attributes of the entities (subject or resource), but also on their behavior using event matching patterns, temporal and cardinality operators defined based on Linear Temporal Logic (LTL) semantics [16]. A policy rule can be defined to restrict access after 3 failed logins, to allow access only after a specific condition is met (e.g., authentication successful), or to trigger a specific compensation action after a specific amount of time. Seckit has been used in different scenarios, such as Cooperative Intelligent Transport Systems (C-ITS) ([17,18]). In this paper the SecKit policy-based framework is used for the specification of policy templates to mitigate privacy threats detected by encrypted traffic analysis.

Other papers proposed the integration of policies with SDN like in [19], where the authors presented an approach to dynamically enforce flow level policies in cloud networks. Policies are written by administrators in high level languages similar to the ones defined above. On the other side, the application of the policies is generic and it is not focused on the mitigation of privacy threats or analysis of encrypted traffic. The authors in [20] proposes a solution in SDN called AnonyFlow, which is an Open-Flow based "in-network" anonymization service with a similar goal to this paper, but where the privacy mitigation is applied only to the anonymization of the IP address. IP addresses are translated to anonymity IDs that only other (i.e., destination) authorized parties can interpret. AnonyFlow is responsible to keep the relation between IDs, hosts or IP addresses. Then, the privacy threat related to the identification of activities with encrypted traffic is not implemented. Another paper with a similar goal to this paper is [21], which focuses on the problem of information disclosure in OpenFlow networks due to the presence of a

scanner or sniffer to understand network patterns. Then, while the objective is similar to this paper, there is no mention in [21] of analysis of encrypted traffic. The authors in [21] use an approach where flow rules are installed by network administrators, but the use of sophisticated policy languages like XACML and SecKit (or MUD discussed in the following sub-section) are not discussed.

2.3. Enforcing Network Restrictions Through the MUD Standard

The Manufacturer Usage Description (MUD) [12] is a recent IETF standard, which defines an architecture and data model to specify network access control constraints, in order to reduce the attack surface in a network. This standard has attracted a significant interest from the research community and standardization organizations, especially because of the potential application in the IoT domain. Indeed, NIST has released periodic reports in the last years about the use of the MUD standard to protect IoT devices against several security attacks. In particular [22], proposes the MUD approach to mitigate distributed threats on home and small business networks, where plug-and-play deployment is required. These aspects were further discussed in a recent report, where potential implementations are described based on a MUD-enabled reference architecture [23]. The use of MUD is also mentioned by [24] to avoid the use of default credentials in IoT devices. Furthermore, NIST has recently published the design of a methodology to generate MUD files using network traffic captures [25].

In general, the research works related to the MUD standard are focused on two main aspects: The obtaining of the MUD file, and the enforcement of the restrictions included in such file. The MUD standard defines a network architecture for obtaining MUD files, and proposes the use of different protocols for the process, such as the Dynamic Host Configuration Protocol (DHCP) [26], the Link Layer Discovery Protocol (LLDP) [27], and X.509 certificates. Additionally, recent works have proposed the use of specific technologies based on the standard architecture. For example, as described by [28], the Cisco MUD Manager implementation (https://github.com/CiscoDevNet/MUD-Manager) makes use of an Authentication, Authorization and Accounting (AAA) infrastructure to obtain the MUD file. Moreover, based on AAA [29], integrates this process with the device's network access phase (i.e., bootstrapping [30]). This way, only legitimate and authorized devices will be able to access the network and deploy their network restrictions. Following a similar approach [31], proposes an extension of the CoAP-EAP approach [32] to obtain MUD files in deployments with resource-constrained devices. In our case, we assume that a MUD file has already been deployed for each end-device in the corresponding network component (e.g., router or switch). Therefore, these approaches are complementary to our proposal, and a potential integration is considered as part of our future work.

Once the MUD file is obtained, the network access control restrictions need to be enforced. While the standard does not define the use of a particular mechanism or technology, most of current approaches make use of software-defined networking (SDN) technology for a dynamic and flexible enforcement approach [31]. For example [33], defines and implements a system for translating MUD rules into flow rules that can be deployed on network switches. Furthermore, the authors define a process for detecting several attacks by using intrusion detection techniques based on such flow rules. As an extension of this work [34], describes the process to detect anomalies by using MUD patterns. Additionally [35], defines the NIST implementation (https://github.com/usnistgov/nist-mud) for the enforcement of MUD restrictions based on OpenFlow-enabled SDN switches [36].

Unlike these approaches, our proposal defines an architecture for modifying MUD rules based on the identification of privacy threats. As shown in the next section, the identification of these threats can trigger the addition of new network restrictions to mitigate such threats based on a policy-based framework. It should be noted that the process for enforcing MUD restrictions could also be based on SDN techniques by considering some of the described works.

2.4. Analysis of Encrypted Traffic

The analysis of network traffic has been the object of an increasing attention by the research community for various objectives. Analysis of network traffic using tailored features or deep learning (DL) algorithms where the features are automatically identified by the algorithm have been presented in literature for a variety of purposes including [37] on the use of DL for service attack detection and analysis of encrypted traffic in [38]. In both cases, high identification is obtained but with a considerable amount of computing resources to execute the DL algorithms. The specific case of analysis of encrypted traffic is used in [39] to improve the efficiency of the network by identifying different types of traffic in the encrypted traffic. On the basis of the results of the analysis of the encrypted traffic, network resources and configurations can be modified. This paper investigates another and more dangerous objective of encrypted traffic analysis: The possibility to infer the activities of a user from the types of traffic identified in the encrypted traffic generated by a user or a set of users. Researchers have demonstrated that even encrypted traffic can reveal the information on specific traffic types like videostreaming web browsing or email [9]. The results from the research community demonstrate that specific types of traffic can be identified not only from the analysis of raw encrypted payload traffic [38,39] (which can be quite demanding from the computing point of view), but also on the basis of derived time based features extracted from the encrypted traffic flows [9], which does not require extensive resources from an attacker. In many cases, these features are available as part of the network monitoring functions and they are easy to access.

Recent surveys on the topic include [8,40], which describe different use cases and techniques of encrypted traffic analysis. In this paper, we propose a machine learning classifier, which identifies different types of traffic to detect potential privacy threats. This approach is based on the ISCXVPN2016 dataset [9], and is further described in the following sections.

Many other papers have investigated the privacy threat due to analysis of encrypted traffic but there are limited works (like this one), which apply privacy mitigation mechanisms based on policies and network profiles. The authors in [41] apply homomorphic encryption to implement packet flow untraceability and message content confidentiality. In particular, authors make use of the Paillier cryptosystem [42], which represents a partially homomorphic cryptosystem that allows to perform sums on encrypted data without having to decrypt first (i.e., additive homomorphism) [43].

The study presented in [44] presents a privacy mitigation technique on encrypted traffic, which is based on a privacy-preserving deep packet filtering protocol. The protocol is composed of a setup phase and a process phase.The setup phase generates encrypted filtering rules, while the process phase matches the encrypted rules with encrypted packets from users. While the goal is similar, the approach is quite different. On the other side, deep packet filtering is known to be quite time consuming to execute and not easy to scale.

Another recent paper [45] with a similar goal to this paper provides another privacy mitigation technique on encrypted traffic. The paper uses an obfuscation technique based on the traffic delay to implement a differentially private (DP) mechanism. The differences with this work is that [45] is applied only to the smart home context while this paper can be applied to any network able to support MUD (e.g., SDN). Then, the technique is applied directly to the traffic, which may not be easy to scale in large networks. Finally, the differentially private (DP) mechanism is relative simple (even if effective in a smart home context) while the framework presented in this paper supports the generation of sophisticated policies.

A recent study on the efficient traffic classification of encrypted and compressed packets is [46] where the authors proposes the randomness evaluation of the payload information in randomly selected network traffic packets. The authors manage to mitigate the risk of the high computing effort of payload inspection by selecting random subsets to enable real-time detection. Even by avoiding the analysis of all the traffic, the authors manage to achieve an excellent classification performance by identifying 94.72% of random packets of 64 Kbytes . The difference of this paper with [46] is that the analysis of encrypted traffic is only one of the components of the proposed approach, while this

paper focuses on the mitigation of the privacy risks once the risk of classifying encrypted traffic is evident. In addition, this paper is based on time-based features extracted from the encrypted traffic while the study in [46] is based on the analysis of the payload traffic. Future developments of this paper may extend the proposed approach based on time-based features to payload analysis by using the methodology described in [46] in particular the random selection of network traffic packets.

Table 1 summarizes the main references identified above and shows which functions are provided by the solutions described in the related reference paper. Even if specific studies manage to support one or maximum two functions, only this paper supports all the functions in a comprehensive framework.

Table 1. Summary of related work and gaps.

Function	References
Use of Policy language to implement sophisticated rules	[10,19–21,34], [This paper]
Mitigation of privacy threats	[20,21,41,44,45], [This paper]
Analysis of encrypted traffic	[8,9,39–41,44–46], [This paper]
Capability to change the network node profile to mitigate attacks	[22,34,36], [This paper]

3. Materials

As discussed before, the proposed framework is composed by three main elements: The policy based framework, the MUD profiles and the machine learning algorithm, which is used to identify potential privacy threats from the encrypted traffic. The aim of this section is to provide an overview of each of these elements, while the description on how these elements are integrated in the overall framework is presented in Section 4. The machine learning element is validated on the ISCXVPN2016 public dataset [9], which stores a large amount of encrypted traffic. This dataset is also described in this section.

3.1. Seckit

In this paper, the Model-based Security Toolkit (SecKit) [10] is adopted as a policy-based framework for specification and enforcement of security policies. Policies in the SecKit framework are specified using an Event-Condition-Action (ECA) rule with the following semantics: Whenever an event is observed and the condition is satisfied, the action is executed. Events are generated by Policy Enforcement Points (PEPs) when an action in a system already took place (actual event), or when an action is about to start (tentative event). For actual events, an ECA rule can be triggered simply to execute a compensation action (detective enforcement) while for tentative event the action part of the ECA rule may dictate to allow, deny, modify, or delay the action that is about to start (preventive enforcement). The responsibility of the PEP is to signal events for which the Policy Decision Point (PDP) has subscribed to, meaning that these events are referenced in the deployed policy rules currently being evaluated.

The condition part of a policy rule may include an expressive set of operators including, in addition to event pattern matching operators (similar to attributes in XACML), also propositional, temporal, and cardinality conditions (see [10] for a complete description). A powerful feature of the SecKit policy language is the definition of policy templates, which are ECA rules parameterized with variables. Using policy templates the same enforcement logic can be applied multiple times to different entities automatically re-using the same template.

Therefore, we use the SecKit policy language for the specification of preventive enforcement policy templates referencing actual events, where compensation actions include the use of MUD profiles, as described in the following subsection. These policy templates act as mitigation actions for the identified privacy threats.

3.2. Manufacturer Usage Description (MUD)

The MUD standard defines a simple architecture for obtaining MUD files using certain technologies such as DHCP or X.509 certificates. In particular, the MUD architecture consists of a device, which sends a MUD URL to indicate where the MUD file is hosted. This MUD URL is sent to the MUD Manager component that requests the file from a MUD File Server, which is associated to a certain manufacturer. In addition to the file itself, the MUD Manager obtains a signature linked with the MUD file to ensure its integrity. Once the file is obtained, the MUD Manager is also responsible for translating the network abstractions of the MUD file into a specific network configuration. Furthermore, this component is intended to maintain network components updated with the appropriate configuration. It should be noted that our proposed architecture considers the use of the MUD Manager for updating the configuration of network components. This update process is driven by the policy-based framework that determines mitigation actions based on the identification of privacy threats. While the process for obtaining the MUD file is out of the scope of this paper, our approach is complementary to the use of specific technologies for that process.

Moreover, the MUD specification defines a data model to describe network access control restrictions, using the Yet Another Next Generation (YANG) [47] and JavaScript Object Notation (JSON) [48] standards. The model is based on the definition of access control lists (ACLs) that restrict communication from/to a certain device. Specifically, a MUD file includes the "mud" container that defines general information about the file, such as "mud-url" (which indicates where the file is hosted), "last-update" (to indicate when the MUD file was generated) or "mud-signature" (which includes a URI to identify the signature associated with a MUD file). In addition, the "mud" container includes the "to-device-policy" and "from-device-policy" containers that indicate the ACLs to be enforced in the communication to/from a device, respectively. These access restrictions are defined by the "acls" container, which augments the model described in [47] by using high-level terms (e.g., "same-manufacturer") to define an expressive and flexible model beyond the use of IP addresses and ports. In our case, as described in Section 5.3, this model is used to define access restrictions for the different traffic types, which are considered in our approach. It should be noted that these restrictions can be updated according to the decisions produced by the policy-based framework to mitigate potential privacy threats.

3.3. Encrypted Traffic Data Set

The approach presented in this manuscript is evaluated using the ISCX VPN-nonVPN traffic dataset ISCXVPN2016 [9], which is briefly described here.

Wireshark and tcpdump were used to collect and analyze the VPN traffic generated using an external VPN service provider, which was connected using OpenVPN in UDP mode.

The authors in [9] have generated different types of traffic, which are identified and described in the following Table 2.

Table 2. Types of traffic and related description.

Type of Traffic	Description
Browsing	HTTPS traffic was generated by users, who were browsing web sites or performing tasks, which included the use of a browser.
Email	This is the traffic type generated by a Thunderbird client. Each client was configured to deliver mail through SMTP/S. The client was configured to receive the traffic using POP3/SSL in one client and IMAP/SSL in the other.
Chat	This traffic type identifies instant messaging applications Skype and pidgin.
Streaming	This traffic type identifies continuous stream of data like the one generated through YouTube or Vimeo services.
File Transfer	This traffic type identifies the traffic generated by applications designed to send or receive files and documents.
VOIP	This is the traffic type generated by voice over ip applications.
P2P	This is the traffic generated by peer to peer and file sharing protocols like Bittorrent.

Overall, the are seven (7) different types of traffic, which are analyzed in this paper. The privacy threat is more relevant if the algorithm for traffic analysis is able to distinguish the each type of traffic from the other types with great accuracy.

The analysis conducted in this paper is based on the Time-Based features extracted from the encrypted traffic rather than the data contained in the payload of the encrypted traffic because the analysis of the Time-Based features is more time efficient. As described in [9], time based features are extracted with different time windows of duration: 15, 30, 60 and 120 s. In this paper, the analysis is performed only on the 15 s duration because the general goal of the framework is to react quickly to the detection of privacy threat and therefore the shortest duration is the most appropriate. In addition, further analysis not presented in this paper has shown that the results for the other time windows are anyway similar to the 15 s duration and for reasons of space, we present the results only for this duration.

In comparison to other studies, where features like Source IP, Destination IP, Source Port, Destination Port and type of Protocol (TCP or UDP) are used, the authors of [9] uses a common definition of flow, where a flow is defined by a sequence of packets with the same values for Source IP, Destination IP, Source Port, Destination Port and type of Protocol (TCP or UDP). While NetMate is often used to generate features from traffic, the authors of [9] have developed an application: ISCXFlowMeter, which generates bidirectional flows where the first packet determines the forward (source to destination) and backward (destination to source) directions. Then the statistical time-related features are calculated separately in the forward and reverse direction. The time related features are calculated using two different approaches: In the first approach, it is calculated the time between packets or the time that a flow remains active. In the second approach, the time is fixed and other variables are measured like bytes per second or packets per second.

The time based features used in this paper are described in [9] and they include the list of 23 features displayed in Table 3. These time based features are used by the SVM machine learning algorithm to discriminate the different types of traffic, which may indicate the privacy threat. The features do not have the same discriminating power: Some features may not reveal the type of traffic while other features introduce the privacy risk. As shown in the Section 5, the obfuscation of the traffic type can be implemented by selecting the least discriminating features, but this information is not known a priori and it must be determined as discussed in Section 5. It is worth noting that the ratio of bytes per packet could add an additional feature as instant messengers do not fill all packages since they exchange sort messages, thus the TCP packet is not expected to be large in content, contrary to data streaming or file transfer. The correlation between feature 22 (Flow bytes per second)

and 23 (Flow packets per second) should be linked to the ratio of bytes per packet and such correlation would be exploited by feature select algorithms like ReliefF used in this paper.

The authors of the public dataset described in [9] have produced two different scenarios: Scenario A with VPN and non-VPN traffic and scenario B where encrypted and VPN traffic are mixed together in one dataset. We have only used the scenario B because it is the most challenging to identify traffic types and it includes only encrypted traffic.

The classification of the encrypted traffic is done using the Support Vector Machine (SVM) machine learning algorithm. Additional details on the machine learning algorithm, metrics and feature selection algorithm are provided in Section 4.3.

Table 3. Time based Features and related IDs.

Id	Description
1	Duration of the flow
2	Forward Inter Arrival Time (FIAT), the time between two packets sent forward direction (mean)
3	FIAT (minimum)
4	FIAT (maximum)
5	FIAT (Standard Deviation)
6	Backward Inter Arrival Time (BIAT), the time between two packets sent backwards (mean)
7	BIAT (minimum)
8	BIAT (maximum)
9	BIAT (Standard Deviation)
10	Flow Inter Arrival Time (FLIAT), the time between two packets sent in either direction (mean)
11	FLIAT (minimum)
12	FLIAT (maximum)
13	FLIAT (Standard Deviation)
14	The amount of time a flow was active before going idle (ATFAI) (mean)
15	ATFAI (minimum)
16	ATFAI (maximum)
17	ATFAI (Standard Deviation)
18	The amount of time a flow was idle before becoming active (ATFIA) (mean)
19	ATFIA (minimum)
20	ATFIA (maximum)
21	ATFIA (Standard Deviation)
22	Flow bytes per second
23	Flow packets per second

4. Architecture and Operation

This section describes the proposed solution where the three components described in the previous Section 3 are integrated in a comprehensive framework. Section 4.1 describes the proposed functional architecture where the relationships among the main components are described. Then, Section 4.2 describes the workflow among the different components and the sequence of operations starting from the analysis of the encrypted traffic by the traffic analysis module, which can detect privacy threats on which the policy based frameworks must take action. Section 4.3 describes the machine learning algorithm (i.e., SVM) used to perform the traffic analysis.

4.1. Functional Architecture

The overall functional architecture for the proposed approach to privacy threats mitigation is shown in Figure 1. This architecture integrates the techniques for identifying privacy threats over encrypted traffic, MUD elements, and different Seckit policy-based components to trigger mitigation actions for such privacy threats. In particular, the architecture consists of the following components:

- End device: It is the component sending and receiving encrypted traffic. An end device could be represented by a laptop, smartphone or other components with network capabilities.
- Network component: It is represented by a switch or router, which is responsible for enforcing the network access restrictions in a MUD file. Such restrictions can be modified according to the identification of privacy threats.
- Traffic Analysis Module: It analyzes the encrypted traffic, which is sent by the network component and extracts time-based features listed in Table 1. These features are communicated through the Time-based Features Interface. This component acts also as a Policy Enforcement Point to change the access to time-based features based on the decisions provided by the Policy Decision Point (PDP).
- Privacy Threat Analysis: It is the component in charge of executing the machine learning algorithm to detect privacy threats. The results of such analysis are sent to the Policy Decision Point to trigger mitigation actions.
- MUD Manager: Its functionality is already defined in the MUD standard [12]. In our case, this component also acts as Policy Enforcement Point to translate potential mitigation actions into specific network configurations to be enforced by network components.
- Policy Decision Point (PDP): It evaluates a set of security policies based on the information being communicated by the Privacy Threat Analysis module. The result of such evaluation is a mitigation action with the objective to limit the access to time-based features or to change network configurations.
- Policy Repository: A database containing a set of security policies. Such policies can be defined by a network administrator or the owner of end devices sending and receiving encrypted traffic.

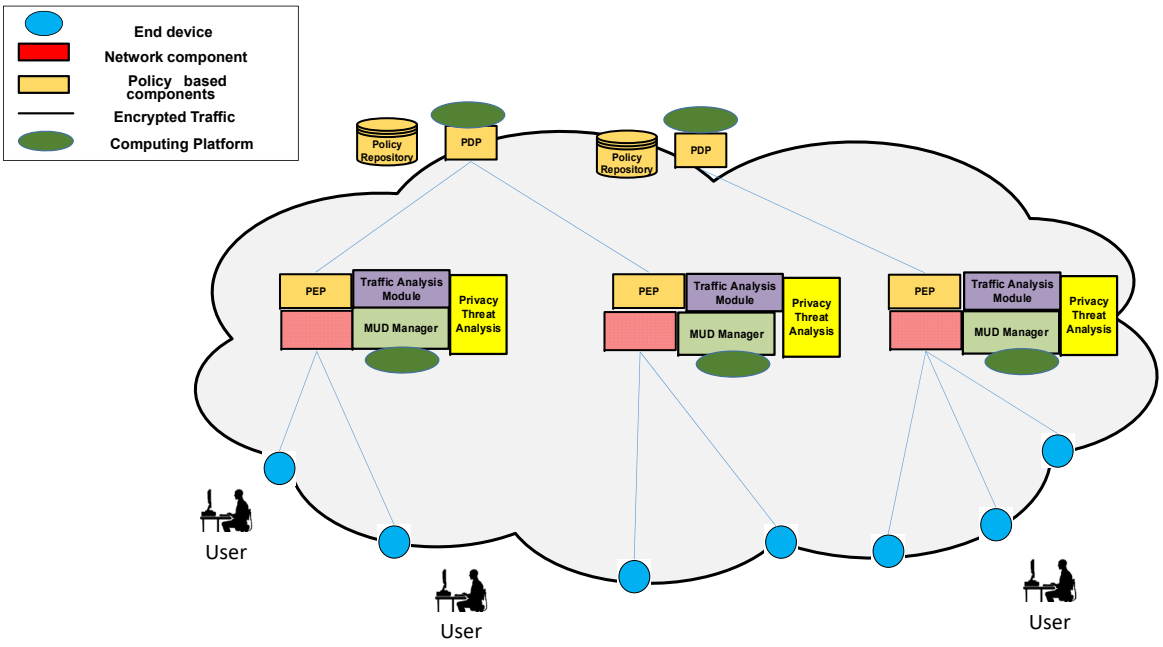


Figure 1. Overall functional architecture for the proposed framework.

The figure also show the computing platforms, which can host the components described above. We note that the proposed components do not impose specific requirements on the computing platforms hosting them and that data replication and load balancing can also be used to meet the demand. Section 5.5 provides a more detailed discussion on the reported performance evaluations from literature for each component.

4.2. Workflow Interactions

Based on the components previously identified, Figure 2 shows the workflow interaction among such components for the identification and mitigation of privacy threats in encrypted traffic. As already mentioned, a network component will be represented by a switch/router, which receives the encrypted network traffic being sent between end devices. In Figure 2 only the Switch/Router is the legacy equipment (i.e., the equipment which already exists in the network before the deployment of the framework proposed in this paper). Then, the encrypted traffic is sent to the Traffic Analysis Module (step 1) to be analyzed. Such analysis is carried out to get the set of time-based features that is described in Table 1. These time-based features are sent to the Privacy Threat Analysis component (step 2), which makes use for the Support Vector Machine (SVM) algorithm for traffic classification (step 3). As described in Section 3.3, we consider three evaluation metrics: Accuracy, precision and recall, which are calculated by considering the number of true/false positives/negatives. After the analysis is complete, the values of such metrics is sent together with the information corresponding with the traffic flow being analyzed to the PDP (step 4). This flow data could make reference to IP addresses or ports associated to the communication that can be used for triggering specific mitigation actions.

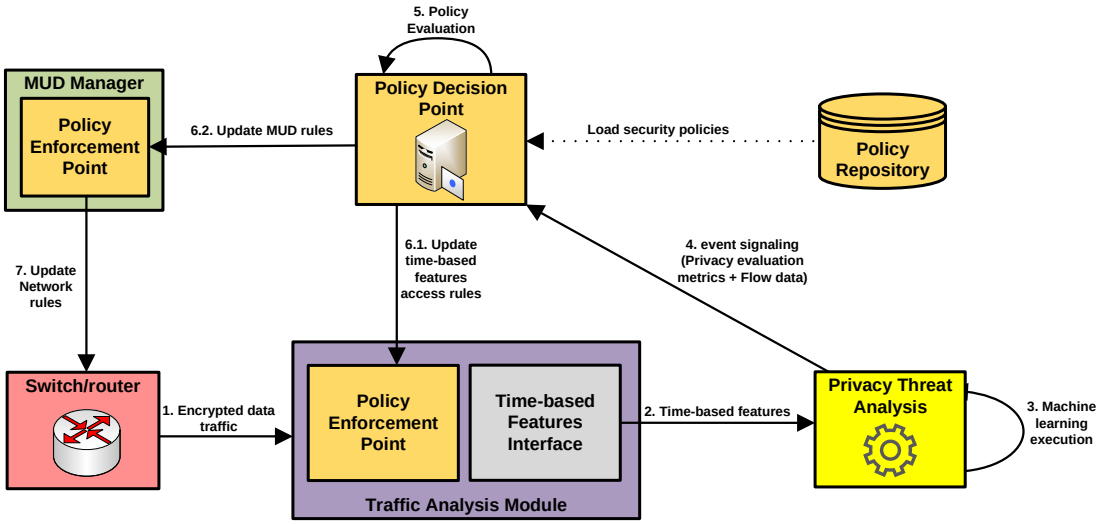


Figure 2. Workflow interactions of the proposed approach to identify and mitigate privacy threats.

Then, the PDP evaluates the policies stored in the Policy Repository (step 5) and generates a mitigation action, which is sent to PEP components depending on the particular action. It should be noted that these policies are based in the Seckit data model, which is adapted to cover the aspects related to privacy threats identification and mitigation. Furthermore, while it is out of the scope of this work, such policies could be defined by a network administrator or end devices' owners depending on the scenario. The definition of these policies is based on Event-Condition-Action (ECA) rules. For example, a policy could indicate IF precision > 0.9, THEN limit_access_to feature1, in which a certain precision value (>0.9) is considered as an indicator that a privacy threat has been detected. In this case, the mitigation action is intended to limit the access to the feature with id = 1 (duration of the flow) in Table 3.

In case the obfuscation or limited access to time-based features are not enough to mitigate the privacy threat, a policy could indicate to restrict the communication with a certain endpoint. Then, depending on the type of mitigation action, a different entity is responsible for enforcing

the action produced by the PDP. If the mitigation action describes the limitation of access to a certain time-based feature, the PEP module of the Traffic Analysis Module is in charge of modifying the access to certain features (step 6.1). In the case that the mitigation action requires additional network restrictions (e.g., to filter the traffic associated to a certain IP address or port), the PDP communicates the action to the MUD Manager (step 6.2). Then the PEP component of this module is responsible for translating the mitigation action into specific network configuration to be enforced by the corresponding switch/router (step 7). In both cases, the results derived from the encrypted traffic analysis are used by the policy-based framework components to trigger and enforce certain mitigation actions.

4.3. Machine Learning

This section describes the machine learning algorithm used to identify the type of traffic. As mentioned before, Support Vector Machine (SVM) was used as machine learning algorithm, but any other machine learning algorithm can be used for the same purpose.

Support Vector Machines were introduced by Vladimir Vapnik and colleagues [49]. The basic idea is that Support Vector Machine classifies data by finding the best hyperplane that separates all data points of one class from those of the other class. SVM was originally defined for binary-classification problems but various techniques have been proposed to extend it to multiclassifier problems as in this case. The best hyperplane for an SVM means the one with the largest margin between the two classes, where margin is the maximal width of the slab parallel to the hyperplane that has no interior data points. SVM is based on the concept of support vectors, which are the data points that are closest to the hyperplane and represent the boundaries.

Because SVM is a binary classifier, the error-correcting output codes (ECOC) model was used for multi-classification. The adopted ECOC uses K(K – 1)/2 binary support vector machine (SVM) models using the one-versus-one coding design, where K is the number of unique class labels.

Radial Basis Function (RBF) was used as a kernel in the SVM algorithm. The RBF kernel was selected after a comparison with the linear and polynomial kernel, which has shown that the classification performance was higher with the RBF kernel. The values of the hyperparameters γ and C factor were identified with an optimization process based on a grid approach with ranges: $\gamma = (2^1...2^{12})$ and $C = (2^1...2^{12})$ with the classification accuracy as metric of evaluation. The values of $\gamma = 2^5$ and $C = 2^9$ provided the highest classification accuracy in the grid.

Three evaluation metrics are used in this paper to evaluate the performance of the proposed algorithms for traffic type detection:

1. Accuracy which is calculated as the sum of true positives (TP) and true negatives (TN) on the overall set of samples.
2. Precision which is calculated as number of true positives (TP) over the sum of true positives (TP) and false positives (FP).
3. Recall, which is calculated as number of true positives (TP) over the sum of true positives (TP) and false negatives (FN).

The classification was performed using a 10-fold approach where the original dataset is randomly partitioned into 10 equal size partial datasets (i.e., folds), where one partial dataset is used for testing and the others nine for training. The results from the 10 folds are averaged to produce the final value of the metrics described above. The value of K = 10 fold was used with K-fold as it was found to result in a model skill estimate with low bias.

The evaluation of the optimal features used for classification was based on the ReliefF feature selection algorithm. The ReliefF algorithm is a filter algorithm, which finds the weights of predictor features in a multiclass categorical dataset. The algorithm penalizes the features that give different values to neighbors of the same class, and rewards features that give different values to neighbors of different classes. Details on the ReliefF algorithm are provided in [50], but the main concept is

that ReliefF first sets all feature weights to 0. Considering a dataset $X = x_1, x_2, ..., x_N$ the algorithm iteratively selects a random observation x_r from the dataset, finds the K-nearest observations to x_r for each class, and updates, for each nearest neighbor x_q, all the weights for the predictor features. In this study, the value of K has been set to 1.

Table 4 summarizes the key parameters used in the application of the machine learning SVM algorithm.

Table 4. Values of the parameters used in the evaluation by the algorithms and methods.

Algorithm/Method	**Parameters**
Support Vector Machine	Kernel=Radial Basis Function(RBF), $\gamma = 2^5$ and $C = 2^9$
K-fold	K = 10
ReliefF algorithm	K = 1

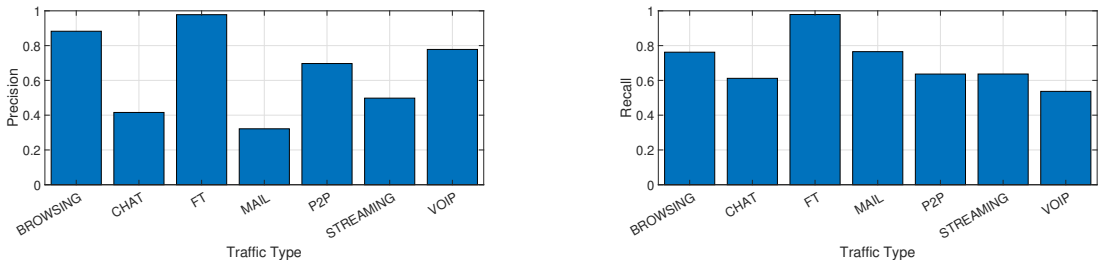
5. Implementation and Evaluation

The aim of this section is to provide a description of a potential implementation of the proposed architecture with examples of the description of the policies, the MUD profiles and the results from the application of machine learning to the analysis of the encrypted traffic before and after the activation of the policies. We describe the results for the encrypted traffic analysis and we provide specific examples of Seckit templates and MUD profiles to mitigate potential privacy threats detected through such analysis. The implementation of each of the specific components is linked among each other as described in Section 4 and in particular the workflow description presented in Figure 2.

In the initial step, an analysis of the traffic is performed to evaluate the possibility to identify privacy vulnerabilities related to the analysis of the encrypted traffic. We note that it is assumed that the attacker cannot access in the network the raw traffic data but only the time based features which are used for network monitoring and which are initially provided by the network to all users. Then, depending on the traffic analysis, the access to such time based features is restricted or not by using the other two components, whose implementation is described in the following sub-sections.

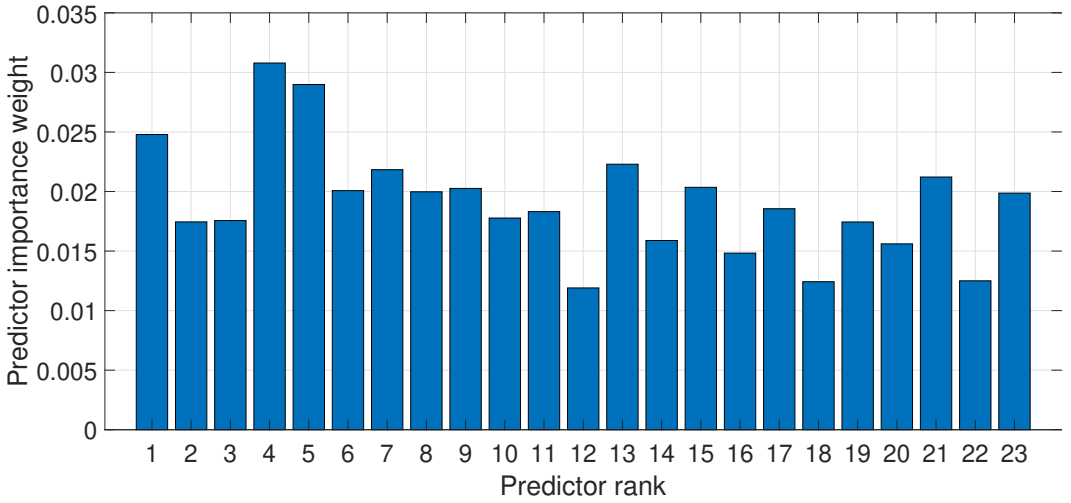
5.1. Analysis of the Encrypted Traffic

Taking into account the approach for encrypted traffic analysis in Section 3.3, Figure 3a,b respectively provide the precision and recall for the identification of the specific traffic types from the encrypted traffic. As it can be seen, it is possible to distinguish in some cases with high accuracy (in particular Browsing, File Transfer and VoIP) the different types of traffic, which may induce a privacy threat. Then, it is important to identify specific features (among the set of 23 features used in the analysis), which need to be obfuscated or removed to decrease identification accuracy and mitigate the privacy threat. This is the reverse problem of increasing the classification accuracy in machine learning problems but the methodology can be quite similar as it based on the analysis and identification of the most discriminating features, which can be performed using different means. As described in the previous sections, we have used ReliefF as a filter feature selection algorithm and we have calculated the classification accuracy of each single feature. The results are provided in Figure 4a for the ReliefF algorithm.

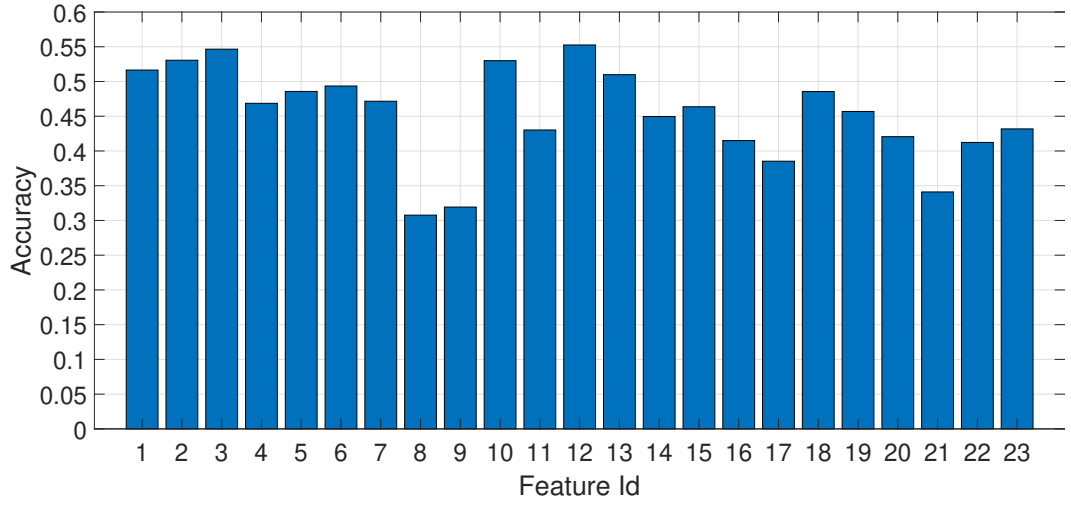


(**a**) (**b**)

Figure 3. Precision and Recall for each type of encrypted traffic when all features are used in absence of privacy mitigation techniques. (**a**) Precision for each type of traffic when all features are used; (**b**) Recall for each type of traffic when all features are used.



(**a**)



(**b**)

Figure 4. Feature selection using ReliefF and assessment of single features; (**a**) Features Ranking using the ReliefF algorithm; (**b**) Accuracy obtained with each single feature.

As it can be seen from the previous figures, some features provides more discriminating power in comparison to others, but the selection of the best features is somewhat not consistent among the two approaches. Then, the worst features (in terms of ranking and classification accuracy) are selected from each of the two approaches and they are compared against the application of all the features.

As described before, three obfuscating techniques are used:

- Use of the features with the lower identification accuracy from Figure 4b. We have used the worst four features. In the following figures, this is called FourWorstFeat.
- Use of the features with lower ranking from the ReliefF algorithm from Figure 4a. We have used the four lowest ranking features. In the following figures, this is called FourLowRankReliefF.
- Addition of Additive White Gaussian Noise (AWGN) on the complete feature set. In the following figures, this is called AllFeaturesNoise. This technique is also adopted in Conditional Privacy to obfuscate information by adding noise [51].

The results for precision and recall of the three techniques in comparison to the application of the entire feature set (AllFeaturesNoNoise) are shown respectively in Figure 5a,b where it is clearly seen that the classification performance drops significantly for each of the used technique. In particular, the traffic types CHAT, MAIL, P2P and STREAMING are basically not distinguishable any longer and the VOIP is significantly degraded. Browsing and FT can be still detected even using low rank or accuracy features. Only the technique based on the addition of noise (i.e., AllFeaturesNoise) is able to degrade the precision and recall also for these types of traffic (i.e., BROWSING and FT).

To show more clearly the impact of the privacy mitigation techniques, Table 5 provides the numeric values corresponding to the Figure 5a,b and the related drop in precision/recall for each of the techniques in comparison to the baseline.

Table 5. Application of the mitigation techniques. Initial values and related drop in precision/recall (in parenthesis) after the application of the mitigation techniques.

Type of Traffic	Baseline with All Features	Four Lowest Ranking Features with Relieff (Precision/Recall Percentage Drop)	Four Features with Lowest Accuracy (Precision/Recall Percentage Drop	Addition of White Gaussian Noise (Precision/Recall Percentage Drop)
Precision				
Browsing	0.8826	0.9104 *(−0.03)*	0.853 *(0.03)*	0.7184 *(0.16)*
Chat	0.4156	0.06 *(0.35)*	0.049 *(0.37)*	0.0134 *(0.40)*
FT	0.9778	0.981 *(0)*	0.979 *(0)*	0.727 *(0.25)*
MAIL	0.321	0 *(0.32)*	0.054 *(0.27)*	0 *(0.32)*
P2P	0.696	0.0386 *(0.66)*	0 *(0.7)*	0 *(0.7)*
STREAMING	0.497	0.079 *(0.42)*	0.207 *(0.29)*	0.147 *(0.35)*
VOIP	0.778	0.503 *(0.27)*	0.176 *(0.60)*	0.280 *(0.50)*
Recall				
Browsing	0.762	0.6 *(0.16)*	0.553 *(0.21)*	0.43 *(0.33)*
Chat	0.611	0.344 *(0.27)*	0.232 *(0.38)*	0.184 *(0.43)*
FT	0.978	0.615 *(0.36)*	0.648 *(0.33)*	0.522 *(0.46)*
MAIL	0.765	0 *(0.77)*	0.625 *(0.14)*	0 *(0.77)*
P2P	0.636	0.397 *(0.24)*	0 *(0.64)*	0 *(0.64)*
STREAMING	0.636	0.381 *(0.26)*	0.326 (0.31)	0.262*(0.37)*
VOIP	0.536	0.49 *(0.05)*	0.355 *(0.18)*	0.36 *(0.18)*

Then, the designer of the mitigation technique can decide to choose and define the specific privacy mitigation techniques in the policies and the MUD profiles according to the type of traffic, which generated the privacy threat. Each technique has its own advantages/disadvantages: AllFeaturesNoise provides all the features but they are strongly obfuscated by noise, while FourWorstFeat and FourLowRankReliefF only provides a subset of features to the public set of users.

The following subsections will describe respectively in Section 5.2 the implementation of the policy based framework to limit the access to the time based features to all the public users and in Section 5.3 the use of the MUD profiles where one of the mitigation techniques described above is not enough to limit the privacy threat and the traffic on a specific node must be denied.

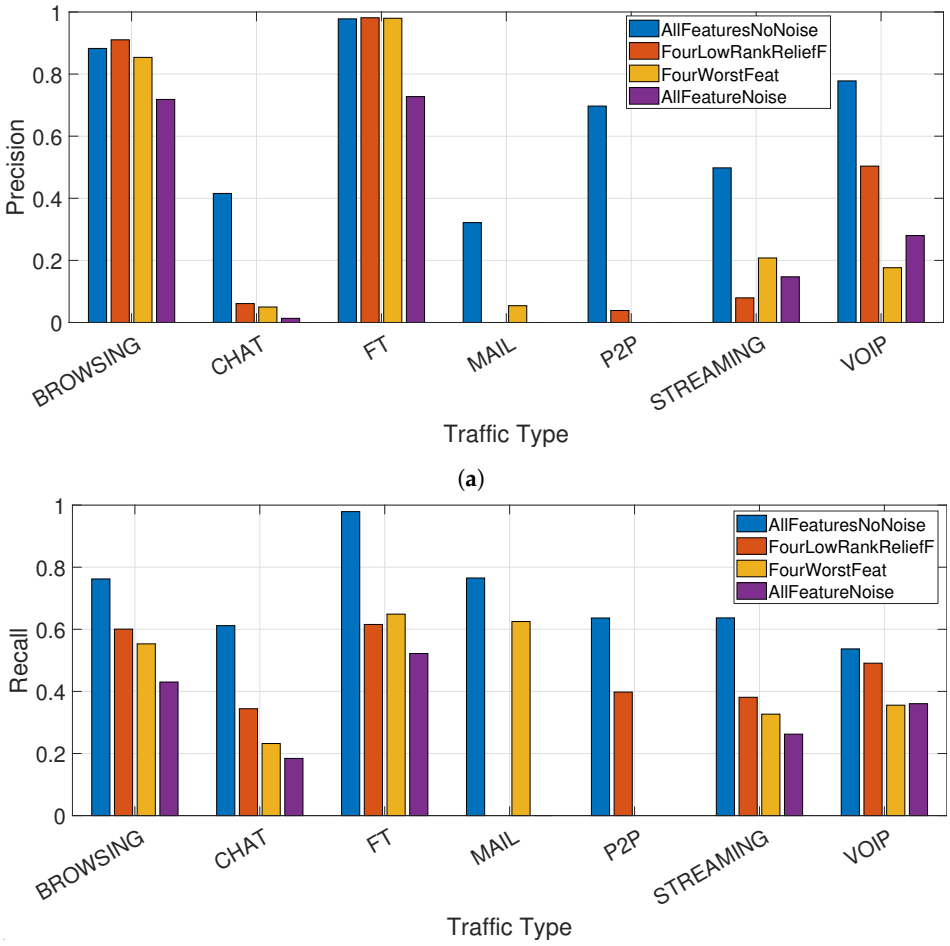


Figure 5. Comparison of the precision and recall between obfuscating techniques and the baseline with all features; (**a**) Comparison of the precision obtained using the different obfuscating techniques and the baseline using all features; (**b**) Comparison of the recall obtained using the different obfuscating techniques and the baseline using all features.

5.2. Description of the Policy Templates

Figure 6 describes two policy templates that were implemented to mitigate the privacy threats in our scenario. The implementation of the policy templates is the subsequent step to the traffic analysis as shown in Figure 2.

The first policy limits the public access to the time based features by using one of the three techniques (FourLowRankReliefF, FourWorstFeat and AllFeatureNoise) described in Section 5.1, when the precision detected in the privacy evaluation metric event is higher than 0.6 and lower or equal to 0.8. The second policy is more restrictive and it uses the MUD profile feature to filter the public access to time based features in case the precision is higher than the threshold value of 0.8. A third policy was also defined for the case when the precision is lower or equal to 0.6, which allows access to all the features and retracts the MUD profile filtering the access, which is the normal operation when a privacy threat has not been detected. Finally, a policy template instantiation rule was also specified, which instantiates the respective policy templates in our implementation for the respective flows.

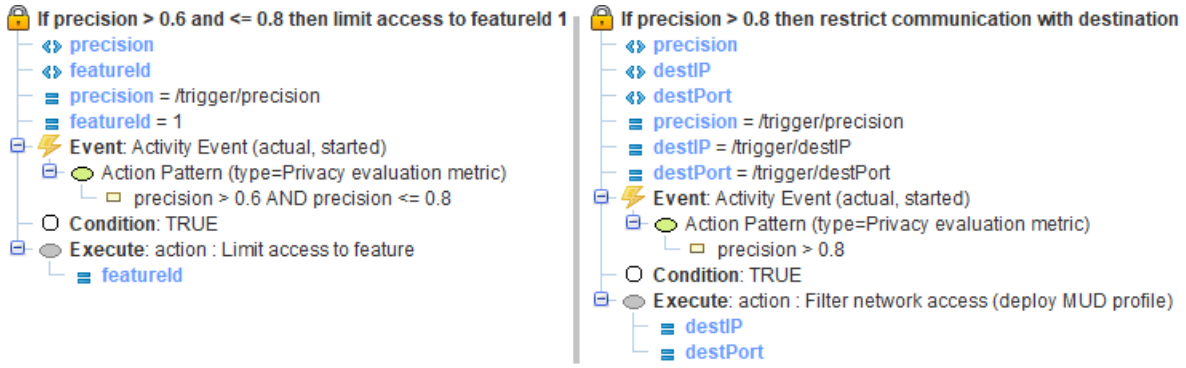


Figure 6. Policy template.

5.3. Description of the MUD Profiles

As described in the previous sub-section, the use of MUD files is considered to restrict the communications of a certain end device through the enforcement of MUD restrictions in a certain network component, when the detection of a traffic activity (and the related privacy threat) cannot be mitigated by the denial of public access to the most discriminating time based features or the application of obfuscation technique with addition of noise. Then, the traffic on one or more nodes must be denied or limited to mitigate the privacy threat. This is an optional step as described in Figure 2. Listing 1 shows an example of MUD file in which the HTTPS and POP3 communications (i.e., browsing and mail traffic) are allowed. This file has been generated by using the tool MUDMaker.

(https://www.mudmaker.org/) and simplified for the sake of clarity. As already mentioned in Section 3.2, a MUD file includes the "mud" and "acls" containers. The former describes different aspects of the MUD file, such as the mud-version, mud-url, mud-signature or last-update, as well as the name of the access control lists, which are defined in the "acls" container. In this case we have defined a single policy ("mud-14561-v4fr") to permit the communication from the end device to certain endpoints through specific ports.

The definition of this policy is included in the "acls" container. In particular, we define two access control entries (ACEs) called loc0-frdev and loc1-frdev with a similar meaning. The former is intended to allow HTTPS traffic by specifying the IPv4 ("type": "ipv4-acl-type") and TCP ("protocol": 6) protocols and port 443 in the field "destination-port". The latter is defined to allow POP3 traffic with the same values and the port 995. Furthermore, both ACEs indicate that this traffic is allowed by using the field "forwarding: Accept" in the "actions" item.

Listing 1: Example of Manufacturer Usage Description (MUD) file to allow browsing and mail communications.

```
{
''ietf-mud:mud'': {
''mud-version'':1,
''mud-url'':''https://manufacturer1/device1.json'',
''mud-signature'': ''https://deviceexamples/ModelName.p7s'',
''last-update'': ''2020-05-08T16:54:50+00:00'',
...
''from-device-policy'': {
''access-lists'': {
''access-list'': [
{''name'': ''mud-14561-v4fr''}
]}},
...
}
''ietf-access-control-list:acls'': {
''acl'': [
''name'': ''mud-14561-v4fr'',
''type'': ''ipv4-acl-type'',
''aces'': {
''ace'': [
{
''name'': ''loc0-frdev'',
''matches'': {
''ietf-mud:mud'': {
''local-networks'': [
null
]
},
''ipv4'': {
''protocol'': 6
},
''tcp'': {
''ietf-mud:direction-initiated'': ''from-device'',
''destination-port'': {
''operator'': ''eq'',
''port'': 443
}
}
},
''actions'': {
''forwarding'': ''accept''
}
},
{
''name'': ''loc1-frdev'',
''matches'': {
''ietf-mud:mud'': {
''local-networks'': [
null
]
},
''ipv4'': {
''protocol'': 6
},
''tcp'': {
''ietf-mud:direction-initiated'': ''from-device'',
''destination-port'': {
''operator'': ''eq'',
''port'': 995
}
}
},
''actions'': {
''forwarding'': ''accept''
}
}]}}}
```

It should be noted that potential extensions for the MUD data model could be considered [31]. For example, another option could be the use of some of the traffic-based features to be included in the MUD file for a more fine-grained representation of data flows. Indeed, these aspects represent part of

our future work in this area. In this paper, the network access control restrictions included in a MUD file are enforced by a network component (e.g., switch or router). Then, based on the policy-based framework decisions, a potential mitigation action is intended to filter the network communication, in case the obfuscation or the limitation to access to time-based features are not enough to mitigate a privacy threat. Following with the previous example, a privacy threat associated to the browsing communication (i.e., HTTPS) could be mitigated by changing the value of the field "forwarding" (i.e., "accept") to "drop" or "reject", by following the data model for the MUD access control lists [47].

5.4. Considerations for the Deployment of the Proposed Approach

This section discusses the aspects related to a potential deployment of the proposed approach. One of the limitations of the proposed approach is that the network operator must implement the components described in this paper. A machine learning computing platform must be implemented to extract the time based features from the traffic flows and identify the possibility of a privacy threat. On the other side, current network infrastructures do already have machine learning components implemented for Intrusion Detection Systems (IDS). While the objectives are different, the classification function can be similar. Obviously the network operator and manager can adopt different machine learning algorithms from the one used in this paper (e.g., SVM). Then, a policy based framework must be implemented. As described in the related work section, various network management policy languages are already defined and implemented, which can fulfill a similar role (even if it may not have the same level of descriptive power of SecKit). Finally, the MUD standard must be implemented in the network infrastructure. The need to implement the three components is one of the main limitations of the proposed approach even if it is mitigated by the considerations above.

One advantage of the proposed approach is that the users do not need to be involved in the definition of the rules in the network unless they want it (to indicate which traffic types are more sensitive). The rules for privacy mitigation (Figure 6 and Listing 1 can be entirely defined by the network manager on the basis of the network topology, node properties and the results of the machine learning output. In particular, the interaction with the machine learning output could be fully automatized by a parameter configurator component (not described in the paper as it is part of future developments) because the classification accuracy results can indicate which actions should be written in the policy and MUD profile definitions. For example, if a traffic type is still detected with high accuracy in the encrypted traffic even with the obfuscation of features or the application of noise, the action for the parameter configurator component is to update the corresponding MUD profile.

Thus, the justification to use the framework proposed in this paper is that it provides a comprehensive set of tools for the network manager to identify privacy threats and mitigate them either with the application of policies which are used to obfuscate the traffic features on one or more nodes or by changing the node profiles to deny traffic. Different obfuscation techniques are used in this paper and the results are compared in Section 5.1, where it can be seen that their application can remove completely the risk of identifying a specific traffic type (e.g., mail or P2P traffic) or reducing it to one third or half (chat or VOIP).

5.5. Evaluation Methodology and Performance Aspects

This paper does not perform an evaluation of the performance of the proposed solution from the scalability point of view but it refers to previous studies where such evaluation is provided. In this subsection, we describe potential metrics of evaluation. The main metric of the evolution is the time requested by the framework to act when a privacy threat is discovered. On this metric, the following considerations can be made. One of the advantages of the proposed framework is that the analysis of the traffic is performed on the time based features instead of the traffic payload, which drastically reduces the analysis time. The performance of the policy based framework has been evaluated in [10], where it was demonstrated that the SecKit implementation is able to handle hundreds of thousands of policy updates/event in seconds using mass-market consumer computers (the evaluation was done in 2015).

Then, this framework is high scalable even for large networks. A performance evaluation of the MUD profile updates is provided in [29], where it can be seen that hundreds of MUD rules can be enforced in seconds. The supported number of MUD rules updates per second is obviously inferior to the policy updates per second, but we have to consider that MUD rules updates are only executed in exceptional cases when the application of a policy with obfuscation of the features fails to reduce significantly the privacy threat.

The computing performance of the SVM machine learning algorithm has also been evaluated and the results are reported in the following Table 6, where the execution time of each of the obfuscation technique is estimated together with the time of the initial operation of calculating the probability of identification of each traffic type. The computing platform used to calculate the values presented in Table 6 is a laptop computer with Intel Core i7 8550U running at a clock speed of 1.8 GHz with 8 GBytes of RAM. For clarity, we repeat here the three obfuscating techniques described above in this section. The use of the lowest ranking features obtained from the application of the feature selection ReliefF algorithm is identified in Table 6 with Application of ReliefF. The application of the 4 features with the lowest single traffic type identification performance is identified in Table 6 with Feature Evaluation. This execution time is much higher than the other operations because the classification process must be repeated for all the 23 time based features. The application of noise to obfuscate the features is identified with Application of noise in Table 6. Note that the total execution time for the application of an obfuscation technique is the combination of the Identification of the traffic types plus the application of the appropriate obfuscation technique and the time needed to select, execute and transmit the policies or the MUD profile information.

Table 6. Execution time for the machine learning component.

Operation	Execution Time
Identification of the traffic types	29.7 s
Application of ReliefF	48.27 s
Feature Evaluation	517 s
Application of noise	30 s

The computers hosting the policy based framework (e.g., PDP) and traffic analysis algorithms can be anyway sized to support the traffic load in the network. In a similar way, they can be designed to be fault tolerant. Each of three components does not impose specific requirements on the design of the computing platform hosting them.

The other metric is related to the mitigation of the privacy threat. This is measured directly with the drop in classification accuracy when a specific mitigation technique is applied. The value of the drop for the three mitigation techniques based on the application of policy is shown in the Section 5.1.

6. Conclusions

The use of well-known security protocols such as Transport Layer Security (TLS) allow most of the Internet traffic to be properly protected by using cryptographic algorithms. However, recent research efforts have demonstrated that even when the communication is encrypted, it is still possible to infer different types of traffic, which can harm users' privacy. To cope with this issue, this paper has described a privacy mitigation framework to address privacy risks related to the analysis of encrypted traffic, which is based on three main components, including a machine learning classifier, a flexible policy-based framework and the application of network node profiles through the MUD standard. A description of the implementation of each of the components and how they are integrated is provided. To evaluate the analysis of the encrypted traffic the public dataset ISCXVPN2016 is used. While the research literature has provided similar solutions for each of the components, no studies (to the knowledge of the authors) has provided a comprehensive set of tools to support the network

managers in a more efficient and effective conformance to privacy regulations in the world including the General Data Protection Regulation (GDPR) in Europe. The justification to use such framework is that it provides a comprehensive set of tools, which can be quite integrated to provide different options for the network manager.

Further developments will investigate improvements of the proposed framework regarding one or more components. Regarding the machine learning component, we will expand the initial analysis of 15 s to 30, 60 and 120 s and a comparison with other machine learning algorithms beyond SVM will be performed. The proposed approach can also be applied to other public datasets of encrypted traffic. Regarding the MUD component, one improvement would be a more refined use of the access control lists in the MUD file to improve the granularity on the control of the traffic data flows. Regarding the policy based component, another improvement is the definition of an automatic parameter configurator suggested in Section 5.4 to automatically generate policies and MUD profiles on the basis of the output of the traffic analysis. Furthermore, we will analyze the integration with SDN-NFV architectures with Fog or Cloud infrastructures to extend the analysis and the application of the privacy mitigation solutions with protocols and standards defined in the Fog and Cloud context.

Author Contributions: Conceptualization, G.B., J.L.H.-R., S.N.; methodology, all authors; software, G.B., J.L.H.-R., R.N.; validation, all authors; investigation, all authors; writing—original draft preparation, G.B. and J.L.H.-R.; writing—review and editing, all authors; supervision, G.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partially supported by the European Commission through project SerIoT funded by the European Union H2020 Programme under Grant Agreement No. 780139. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission.

Acknowledgments: We acknowledge the reviewers for taking their time to review the manuscript and to improve the quality of the publication.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AAA	Authentication, Authorization and Accounting
ACE	Access Control Entry
ACL	Access Control List
AWGN	Additive White Gaussian Noise
BIAT	Backward Inter Arrival Time
C-ITS	Cooperative Intelligent Transport Systems
CoAP	Constrained Application Protocol
DHCP	Dynamic Host Configuration Protocol
EAP	Extensible Authentication Protocol
ECA	Event-Condition-Action
ENISA	European Union Agency for Cybersecurity
FIAT	Forward Inter Arrival Time
FLIAT	Flow Inter Arrival Time
HTTPS	HyperText Transfer Protocol Secure
IETF	Internet Engineering Task Force
IPSec	IP Security
JSON	JavaScript Object Notation
LLDP	Link Layer Discovery Protocol
LTL	Linear Temporal Logic
MUD	Manufacturer Usage Description
NGAC	Next Generation Access Control
P2P	Peer-to-Peer
PDP	Policy Decision Point
PEP	Policy Enforcement Point

SDN	Software-Defined Networking
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
VPN	Virtual Private Network
VoIP	Voice over IP
XACML	eXtensible Access Control Markup Language
YANG	Yet Another Next Generation

References

1. Rescorla, E. Rfc 8446: The Transport Layer Security (TLS) Protocol version 1.3. Internet Eng. Task Force (IETF) 2018. ISSN 2070-1721. Available online: https://tools.ietf.org/html/rfc8446 (accessed on 18 September 2020).
2. Kent, S.; Seo, K. IETF RFC 4301: Security Architecture for the Internet Protocol. 2005. Available online: https://tools.ietf.org/html/rfc4301 (accessed on 18 September 2020).
3. Google. HTTPS Encryption on the Web. 2020. Available online: https://transparencyreport.google.com/https/overview (accessed on 18 September 2020).
4. Apthorpe, N.; Reisman, D.; Feamster, N. A smart home is no castle: Privacy vulnerabilities of encrypted IoT traffic. *arXiv* **2017**, arXiv:1705.06805.
5. Sherry, J.; Lan, C.; Popa, R.A.; Ratnasamy, S. Blindbox: Deep packet inspection over encrypted traffic. In Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, London, UK, 17–21 August 2015; pp. 213–226.
6. Bissias, G.D.; Liberatore, M.; Jensen, D.; Levine, B.N. Privacy vulnerabilities in encrypted HTTP sPrivacy vulnerabilities in encrypted HTTP streamstreams. In *International Workshop on Privacy Enhancing Technologies*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 1–11.
7. Sun, Q.; Simon, D.R.; Wang, Y.M.; Russell, W.; Padmanabhan, V.N.; Qiu, L. Statistical identification of encrypted web browsing traffic. In Proceedings of the 2002 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 12–15 May 2002; pp. 19–30.
8. Paraskevi, D.; Fajfer, J.; Müller, N.; Papadogiannaki, E.; Rekleitis, E.; Střasák, F. Encrypted Traffic Analysis, ENISA Report. 2019. Available online: https://www.enisa.europa.eu/publications/encrypted-traffic-analysis (accessed on 18 September 2020).
9. Draper-Gil, G.; Lashkari, A.H.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of encrypted and vpn traffic using time-related features. In Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP), Rome, Italy, 19–21 February 2016; pp. 407–414.
10. Neisse, R.; Steri, G.; Fovino, I.N.; Baldini, G. SecKit: A model-based security toolkit for the internet of things. *Comput. Secur.* **2015**, *54*, 60–76. [CrossRef]
11. OASIS. eXtensible Access Control Markup Language Version 3.0. 2013. Available online: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html (accessed on 18 September 2020).
12. Lear, E.; Romascanu, D.; Droms, R. Manufacturer Usage Description Specification (RFC 8520). 2019. Available online: https://datatracker.ietf.org/doc/rfc8520/ (accessed on 18 September 2020).
13. Alsmadi, I.; Xu, D. Security of software defined networks: A survey. *Comput. Secur.* **2015**, *53*, 79–108. [CrossRef]
14. Atlam, H.F.; Alassafi, M.O.; Alenezi, A.; Walters, R.J.; Wills, G.B. XACML for Building Access Control Policies in Internet of Things. In Proceedings of the IoTBDS, Funchal/Madeira, Portugal, 19–21 March 2018; pp. 253–260.
15. Ferraiolo, D.; Chandramouli, R.; Kuhn, R.; Hu, V. Extensible access control markup language (XACML) and next generation access control (NGAC). In Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control, New Orleans, LA, USA, 11 March 2016; pp. 13–24.

16. Gerth, R.; Peled, D.; Vardi, M.Y.; Wolper, P. Simple on-the-fly automatic verification of linear temporal logic. In *Proceedings of the International Conference on Protocol Specification, Testing and Verification*; Springer: Berlin/Heidelberg, Germany, June 1995; pp. 3–18.
17. Baldini, G.; Neisse, R. On the application of Policy-based Frameworks to Autonomous Vehicles. In Proceedings of the 2020 IEEE Global Internet of Things Summit (GIoTS), Dublin, Ireland, 3 June 2020; pp. 1–6.
18. Hernández-Ramos, J.L.; Baldini, G.; Neisse, R.; Al-Naday, M.; Reed, M.J. A Policy-based Framework in Fog enabled Internet of Things for Cooperative ITS. In Proceedings of the 2019 IEEE Global IoT Summit (GIoTS), Aarhus, Denmark, 17–21 June 2019; pp. 1–6.
19. Bellessa, J.; Kroske, E.; Farivar, R.; Montanari, M.; Larson, K.; Campbell, R.H. Netodessa: Dynamic policy enforcement in cloud networks. In Proceedings of the 2011 IEEE 30th Symposium on Reliable Distributed Systems Workshops, Madrid, Spain, 4–7 October 2011; pp. 57–61.
20. Mendonca, M.; Seetharaman, S.; Obraczka, K. A flexible in-network IP anonymization service. In Proceedings of the 2012 IEEE international conference on communications (ICC), Ottawa, ON, Canada, 10–15 June 2012; pp. 6651–6656.
21. Klöti, R.; Kotronis, V.; Smith, P. OpenFlow: A security analysis. In Proceedings of the 2013 21st IEEE International Conference on Network Protocols (ICNP), Goettingen, Germany, 7–10 October 2013; pp. 1–6.
22. Polk, T.; Souppaya, M.; Barker, W.C. Mitigating IoT-Based Automated Distributed Threat. 2017. Available online: https://csrc.nist.gov/publications/detail/white-paper/2017/10/12/mitigating-iot-based-automated-distributed-threats/draft (accessed on 18 September 2020)
23. NIST. Securing Small-Business and Home Internet of Things Devices: NIST SP 1800-15. 2019. Available online: https://csrc.nist.gov/publications/detail/sp/1800-15/draft (accessed on 18 September 2020).
24. Jeffrey, V.; Rick, K.; Phillip, L.; Sophia, A. NISTIR 8222: Internet of Things (IoT) Trust Concerns. 2018. Available online: https://csrc.nist.gov/publications/detail/white-paper/2018/10/17/iot-trust-concerns/draft (accessed on 18 September 2020).
25. Watrobski, P.; Klosterman, J.; Barker, W.; Souppaya, M. Methodology for Characterizing Network Behavior of Internet of Things Devices (Draft). Technical Report. Available online: https://csrc.nist.gov/publications/detail/white-paper/2020/04/01/methodology-for-characterizing-network-behavior-of-iot-devices/draft (accessed on 18 September 2020).
26. Droms, R. Dynamic Host Configuration Protocol (RFC 2131). 1997. Available online: https://tools.ietf.org/html/rfc2131 (accessed on 18 September 2020).
27. Link Layer Discovery Protocol. Available online: https://en.wikipedia.org/wiki/Link_Layer_Discovery_Protocol (accessed on 18 September 2020).
28. Feraudo, A.; Yadav, P.; Mortier, R.; Bellavista, P.; Crowcroft, J. SoK: Beyond IoT MUD Deployments–Challenges and Future Directions. *arXiv* **2020**, arXiv:2004.08003.
29. García, S.N.M.; Molina Zarca, A.; Hernández-Ramos, J.L.; Bernabé, J.B.; Gómez, A.S. Enforcing Behavioral Profiles through Software-Defined Networks in the Industrial Internet of Things. *Appl. Sci.* **2019**, *9*, 4576. [CrossRef]
30. Sarikaya, B.; Sethi, M.; Garcia-Carillo, D. Secure IoT Bootstrapping: A Survey, 2018. Available online: https://tools.ietf.org/id/draft-sarikaya-t2trg-sbootstrapping-05.html (accessed on 18 September 2020).
31. Matheu, S.N.; Robles Enciso, A.; Molina Zarca, A.; Garcia-Carrillo, D.; Hernández-Ramos, J.L.; Bernal Bernabe, J.; Skarmeta, A.F. Security Architecture for Defining and Enforcing Security Profiles in DLT/SDN-Based IoT Systems. *Sensors* **2020**, *20*, 1882. [CrossRef] [PubMed]
32. Garcia-Carrillo, D.; Marin-Lopez, R. Multihop bootstrapping with EAP through CoAP intermediaries for IoT. *IEEE Internet Things J.* **2018**, *5*, 4003–4017. [CrossRef]
33. Hamza, A.; Gharakheili, H.H.; Sivaraman, V. Combining MUD policies with SDN for IoT intrusion detection. In Proceedings of the 2018 Workshop on IoT Security and Privacy, Budapest, Hungary, 20 August 2018; pp. 1–7.
34. Hamza, A.; Gharakheili, H.H.; Benson, T.A.; Sivaraman, V. Detecting volumetric attacks on loT devices via SDN-based monitoring of MUD activity. In Proceedings of the 2019 ACM Symposium on SDN Research, San Jose, CA, USA, 3–4 April 2019; pp. 36–48.
35. Ranganathan, M. Soft MUD: Implementing manufacturer usage descriptions on OpenFlow SDN switches. In Proceedings of the International Conference on Networks (ICN), Valencia, Spain, 24–28 March 2019.

36. McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J. OpenFlow-enabling innovation in campus networks. *ACM Sigcomm Comput. Commun. Rev.* **2008**, *38*, 69. [CrossRef]
37. Catak, F.O.; Mustacoglu, A.F. Distributed denial of service attack detection using autoencoder and deep neural networks. *J. Intell. Fuzzy Syst.* **2019**, *37*, 3969–3979. [CrossRef]
38. Wang, W.; Zhu, M.; Wang, J.; Zeng, X.; Yang, Z. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, China, 22–24 July 2017; pp. 43–48. [CrossRef]
39. Wang, P.; Ye, F.; Chen, X.; Qian, Y. Datanet: Deep learning based encrypted network traffic classification in sdn home gateway. *IEEE Access* **2018**, *6*, 55380–55391. [CrossRef]
40. Velan, P.; Čermák, M.; Čeleda, P.; Drašar, M. A survey of methods for encrypted traffic classification and analysis. *Int. J. Netw. Manag.* **2015**, *25*, 355–374. [CrossRef]
41. Fan, Y.; Jiang, Y.; Zhu, H.; Shen, X. An efficient privacy-preserving scheme against traffic analysis attacks in network coding. In Proceedings of the IEEE INFOCOM 2009, Rio de Janeiro, Brazil, 19–25 April 2009; pp. 2213–2221.
42. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 223–238.
43. Jost, C.; Lam, H.; Maximov, A.; Smeets, B.J. Encryption Performance Improvements of the Paillier Cryptosystem. *IACR Cryptol. EPrint Arch* **2015**, *2015*, 864.
44. Lin, Y.H.; Shen, S.H.; Yang, M.H.; Yang, D.N.; Chen, W.T. Privacy-preserving deep packet filtering over encrypted traffic in software-defined networks. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–7.
45. Liu, J.; Zhang, C.; Fang, Y. Epic: A differential privacy framework to defend smart homes against internet traffic analysis. *IEEE Internet Things J.* **2018**, *5*, 1206–1217. [CrossRef]
46. Casino, F.; Choo, K.K.R.; Patsakis, C. Hedge: Efficient traffic classification of encrypted and compressed packets. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2916–2926. [CrossRef]
47. Jethanandani, M.; Blair, D.; Huang, L.; Agarwal, S. YANG Data Model for Network Access Control Lists (RFC8519). 2019. Available online: https://tools.ietf.org/html/rfc8519 (accessed on 18 September 2020).
48. Bray, T. The JavaScript Object Notation (JSON) Data Interchange Format (RFC8259), 2017. Available online: https://tools.ietf.org/id/draft-ietf-jsonbis-rfc7159bis-04.html (accessed on 18 September 2020).
49. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
50. Robnik-Šikonja, M.; Kononenko, I. Theoretical and empirical analysis of ReliefF and RReliefF. *Mach. Learn.* **2003**, *53*, 23–69. [CrossRef]
51. Agrawal, R.; Srikant, R. Privacy-preserving data mining. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 16–18 May 2000; pp. 439–450.