future internet

MDPI

Article

Security Awareness in Software-Defined Multi-Domain 5G Networks

Jani Suomalainen *, Kimmo Ahola, Mikko Majanen, Olli Mämmelä and Pekka Ruuska

VTT Technical Research Centre of Finland, Tietotie 3, 02044 Espoo, Finland; Kimmo.Ahola@vtt.fi (K.A.); mikko.majanen@vtt.fi (M.M.); olli.mammela@vtt.fi (O.M.); Pekka.Ruuska@vtt.fi (P.R.)

* Correspondence: jani.suomalainen@vtt.fi

Received: 28 December 2017; Accepted: 6 March 2018; Published: 8 March 2018

Abstract: Fifth generation (5G) technologies will boost the capacity and ease the management of mobile networks. Emerging virtualization and softwarization technologies enable more flexible customization of network services and facilitate cooperation between different actors. However, solutions are needed to enable users, operators, and service providers to gain an up-to-date awareness of the security and trustworthiness of 5G systems. We describe a novel framework and enablers for security monitoring, inferencing, and trust measuring. The framework leverages software-defined networking and big data technologies to customize monitoring for different applications. We present an approach for sharing security measurements across administrative domains. We describe scenarios where the correlation of multi-domain information improves the accuracy of security measures with respect to two threats: end-user location tracking and Internet of things (IoT) authentication storms. We explore the security characteristics of data flows in software networks dedicated to different applications with a mobile network testbed.

Keywords: 5G; mobile network; security; monitoring; multi-domain; SDN

1. Introduction

Fifth generation (5G) standards are opening mobile networks to new kinds of applications, devices, and business actors [1,2]. In the future, 5G standards will support heterogeneous access methods with high-speed, low-latency, and high-availability, and optimize mobile networks, e.g., for safety-critical (traffic, ehealth) and massive-scale (Internet of things—IoT) applications. Fifth generation systems exploit software networking and network virtualization technologies to enable new stakeholders such as virtual operators, infrastructure providers, or third-party service providers to cooperate more easily and create end-to-end services. In particular, 5G standards will address network management challenges and ease the deployment of secure services over complex multi-domain, multi-operator, and multi-technology networking environments. Fifth generation standards will also ease the customization of network services to fulfil application-specific security needs. In 5G systems, end-to-end application-layer security can be comprised of more than just the secure tunnels (HTTPS, TLS, or VPN) used in previous generations. Thus, 5G operators will be able to provide efficient security and availability guarantees, e.g., with monitoring and access control solutions that are customized for applications. On the other hand, new business and industrial applications will make mobile networks tempting to adversaries and the new technologies will introduce quite different vulnerabilities and threats.

Security monitoring in fourth generation (4G) mobile networks is domain- and operator-specific. Operators monitor their networks for intrusions with different proprietary solutions from telecom infrastructure providers. Monitored information is not shared across administrative borders due to both business reasons and 4G specifications, which essentially have not supported the exchange of trust or security monitoring information. Hence, users and service providers have to trust or distrust operators completely

without a means to evaluate the network's trustworthiness. In 5G systems, with more cooperating parties and diversified service levels, such unquantified trust relationships are not adequate.

Users and service providers need an up-to-date awareness of the security and trustworthiness of 5G systems. However, assessing and measuring the overall security of end-to-end 5G systems is demanding, as monitoring systems must collect and combine information from different sources operated by distrusting actors. In addition, the increased traffic volumes, heterogeneous access technologies, and device types will cause resourcing challenges. Consequently, monitoring systems must be flexible and scalable. They must be flexible since they must ease interoperability and enable the sharing and the correlation of monitored information from several domains in order to form complete end-to-end trust awareness. They must be scalable as they should enable various algorithms to monitor large volumes of heterogeneous traffic streams flowing through mobile networks.

We contribute by presenting how existing technologies can be practically combined for fine-grained, customized, and extensive security monitoring. We show how a various actors and different technologies can cooperate in order to extract knowledge of the complex 5G security landscape. In this paper, we propose a flexible and scalable framework for security monitoring, machine learning, trust measuring and security control in 5G networks. The framework enables different parties—end-users, application providers, third party service providers, and customer organizations, as well as (cooperating) operators and infrastructure providers—to evaluate the trustworthiness of a mobile network. We propose a Trust Level Agreement mechanism for sharing real-time security awareness in multi-domain/multi-operator scenarios. We also describe approaches to address two 5G threats: tracking of end-users' location and authentication storms caused by IoT botnets. Our monitoring approach is evaluated using a mobile network testbed and some selected machine learning algorithms for anomaly detection.

The framework leverages Software-Defined Network (SDN) and Network Function Virtualization (NFV) concepts as well as big data processing and analytic engines (such as Apache Kafka and Spark) to increase the accuracy, flexibility, and scalability of monitoring. Our hypothesis is that the framework provides, when compared to monitoring of heterogeneous data flows combining traffic from various 4G applications, more customized and accurate information of networks' trust levels to operators and user organizations in a real-time mode. With software networks, we isolate application-specific data streams and thus enable the monitoring system to focus on specific traffic patterns that are homogeneous. Thus, the security monitoring system can apply machine learning algorithms and control strategies that are optimal for the application. In addition, by correlating security event information from various 5G domains, we are able to capture threats that were previously hard to address. Virtualization and sharing of security information enable service providers to extend their monitoring landscape to the 5G systems. Hence, security awareness improves, and mitigations speed-up and automatize when application providers receive security information from operators' and infrastructure providers' domains. Figure 1 illustrates our contributions and how the monitoring landscape will evolve when entering the 5G era. The figure illustrates the shift from the generic security monitoring approaches of 4G to customized monitoring and cross-domain sharing of trust relevant information in virtualized 5G networks.

The rest of this paper is structured as follows. First, in Section 2, we survey related work on 5G security monitoring and explain how our contributions map to the 5G security landscape. We also classify security and trust parameters shared over multiple domains and give an overview of software networking technologies that our work builds on. In Section 3, we describe requirements for the monitoring framework and enablers. Section 4 discusses two specific threat scenarios as well as mechanisms for detection and mitigation. It illustrates the advantages of multi-domain data sharing and correlation analysis. Section 5 describes our implementations and technology choices for the framework as well as the testbed. Section 6 studies characteristics of data in SDNs dedicated for different applications. In particular, we illustrate and study how homogeneity of data affects machine learning based on a clustering algorithm. Section 7 discusses the applicability and limitations of the proposed approach and, finally, Section 8 concludes the paper.

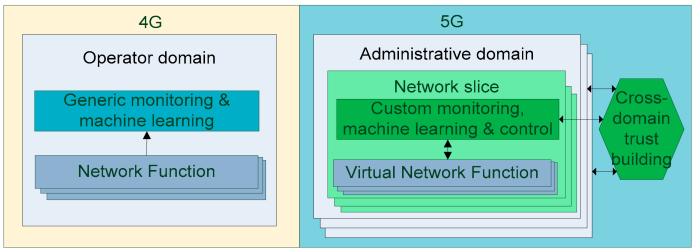

Figure 1. Evolution of security monitoring from fourth generation (4G) to fifth generation (5G) of mobile networks. The foci of our investigation—software-defined and big data analytic means to customize monitoring, machine learning, and control of virtualized 5G networks as well as means to share trust-relevant knowledge across domains—are highlighted in green.

2. Related Work

This section provides background on 5G security and surveys existing research and standardization efforts related to monitoring and softwarization of mobile networks.

2.1. Fifth Generation Security

Fifth generation standards are advancing mobile communication systems in several areas. For the end-users, the evolution of access interfaces will advance the service quality—with increased bandwidth, lower latency, higher availability—and make the connectivity available to billions of new kinds of devices and applications, including IoT [3]. For operators, 5G systems provide lower operating expenses, as new management approaches and virtualization technologies will improve cooperation between different parties and ease service customization.

Fifth generation security architecture is an evolution of third and fourth generation (3G/4G) security standards and architecture defined by Third Generation Partnership Project (3GPP) [4–8]. These specifications focus on the authentication between subscriber's terminals and networks as well as on protection of access and core network communication against external threats. Fifth generation security approaches address these requirements but also new threats arising from new technologies and applications. The requirements for security mechanisms include:

- Scalability and performance—Solutions for new and legacy threats must support high traffic volumes and large device numbers with new emerging applications. Security solutions must be scalable in a horizontal manner (more connected devices) and in a vertical manner (support different applications with diverse requirements).
- Insider threat protection—Increased cooperation between different parties as well as complicated security demands of emerging applications are new challenges. They force 5G security to focus more on threats, which originate e.g., from weakly protected partner networks or from mobile botnets [9,10].

The security architecture provides a holistic security perspective to mechanisms and interactions in the 5G systems. Figure 2 illustrates a domain model of the security architecture [11,12] that was developed in the 5G-ENSURE project. It illustrates domains with trust relationships. The model divides 5G horizontally into infrastructure and tenant domains—due to emerging virtualization of devices and hardware—and vertically into user equipment, operator, and external network domains. These domains are divided, respectively, into user subscriber identity module (USIM), identity management (IM), and mobile equipment (ME) domains; to access network with different Radio Access Technologies (RATs), serving network, transport network, and home network domains; and to third party and Internet Protocol (IP) service provider domains. Slice domains illustrate the potential of 5G for customization—dedicating separate logical resources for different applications or users. Management

domains, where security monitoring solutions lie, are presented separately to emphasize 5G's focus on efficiency and operability of networks.

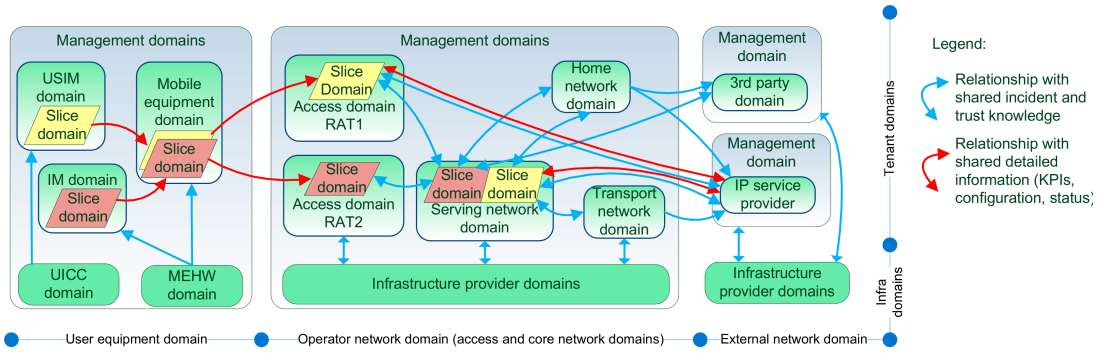

Figure 2. Fifth generation (5G) security architecture (adapted from [11,12]) reflecting security monitoring information that is shared across domains. USIM: user subscriber identity module; RAT: Radio Access Technology; IM: identity management; KPI: key performance indicator; MEHW: mobile equipment hardware; UICC: universal integrated-circuit card; IP: internet protocol.

2.2. Security Monitoring and Trust Management

Security monitoring is a process of collecting, analysing, and inferring security event information in order to gain awareness of a system's security state and trustworthiness as well as to detect and enable responses to security incidents. Monitoring systems collect and share information on events—occurrences that are relevant to the security of the system—from various sources. Networks produce large amounts of event data. Security relevant information is composed of the following categories:

1. *Network configuration* information reveals security capabilities and trustworthiness of the hardware and software deployed to the network. This information consists of software configuration of network functions (software vendors, identities, version information, and management practices) and physical configuration of the infrastructure (topology, location of nodes, physical security, hardware vendors, models). Furthermore, the number, location, and configuration of end-user devices, affect the system's security.
2. *Status information* on available security services allows for keeping track of protected assets. For instance, use of different security protocols, algorithms, firewalls, and secure tunnelling solutions should be monitored and failures recorded.
3. *Traffic statistics*, various counters, and key performance indicators (KPIs) can be used to detect different ongoing threats and attack situations. For instance, abrupt high traffic peaks or resource starvation situations can indicate malicious attacks.
4. *Application data* flowing through the network can be analysed in detail when packet traces (header and payload data from different protocol layers) are available. Packets can be scanned, e.g., for malware. However, in many cases encryption prevents such inspections anywhere other than in the originating and destination domains.
5. *Detected incidents* reports are shared across domains to enable common defence actions e.g., against distributed-denial-of-service attacks.

Different domains in the security architecture (Figure 2) all produce detailed data from these categories. When sharing information across administrative boundaries (between operator and external network domains) the focus has been on incident reporting that enable common reactions to ongoing attacks such as distributed denial-of-service. As sharing of monitoring data in large volumes is unfeasible and as this data may contain privacy- or business-critical information, security

analyses are typically performed locally within administrative domains. However, there is a need for cross-domain sharing of coarse-grained data on network trust compliance. Further, sharing of more detailed information on configuration, statuses, and KPIs is beneficial when it enables mitigations of more subtle threats.

Security monitoring is not a specific theme in 3GPP. However, 3GPP has specified [13,14] hundreds of counters and KPIs to monitor, assess, and optimize mobile networks' operability and efficiency. These KPIs include parameters related to the performance of radio interface, network services, and user's experience. Some of these counters and KPIs also function as security indicators. Actually, all abnormal or unsuccessful incidents in network systems, such as failed handovers or connection setups, are by default carefully counted, and the reasons for failures are recorded whenever possible. Furthermore, all major mobile system vendors (Nokia, Ericsson, Huawei and others) have extended the standard approach and they provide advanced software products for network performance and security monitoring as well as management.

The European Telecommunications Standards Institute (ETSI) NFV Industry Specification Group is addressing the monitoring and management of mobile networks based on virtualization technologies. The group has defined [15] scenarios and requirements for monitoring and discussed related orchestration and management issues. The group has also published security and trust guidelines [16], illustrating use cases and trust issues between different domains in virtualized future telecommunication architectures. They consider the life cycle of trust (evaluating, re-evaluating, invalidating, re-establishing) and present different methods for trust evaluation—including reputational, game theoretical, probabilistic, and look-up tables. The guidelines classify trust relevant parameters in to the following categories: geographical or logical network location, jurisdiction, hardware and software capabilities, origin, ownership, execution history, last audit, and chain of trust, and security mechanisms (encryption, physical security etc.).

The European Union Agency for Network and Information Security (ENISA) has surveyed threats in the 5G landscape [17]. Their recommendations include the development of incident response capabilities and information sharing practices among operators. Cooperation between operators requires both capable computer emergency response teams and organizations enabling cooperation as well as technical mechanisms for sharing of real-time information.

Research efforts for security management approaches for mobile networks include, e.g., Yan et al. [18], who defined a high-level NFV-based framework and requirements for adaptive 5G security management. Luo et al. [19] proposed security assessment mechanisms for SDN-based 5G networks. The mechanism utilized attack graphs and an analytic hierarchy process to quantify security levels as well as to evaluate costs of attacks and security.

Machine learning approaches [20] for monitoring the security state of mobile networks have also drawn academic interest. Gupta et al. [21] used a supervised learning algorithm to analyse IP packet streams from mobile terminals in order to detect distributed denial-of-service attacks. Shabtai et al. [22] analysed mobile network traffic flows to detect malware.

Situational awareness solutions monitor a network in order to make automated decisions based on analysed context knowledge. For 5G, Marquezan et al. [23] studied a monitoring and security adaptation of 5G radio and access networks. Lopez et al. [24] proposed an incident management architecture that combines different layers of 5G information, including NFV and SDN aspects, conventional risk management schemes, and control loop for adaptive security.

2.3. Security by Software-Defined Networking

Software-Defined Networks (SDNs) ease network configuration and evolution as well as policy enforcement. SDN is based on three principles that enable faster provisioning and configuration of network connections [25–27]:

1. Decoupling of control and data plane—Data plane nodes (switches) query the control plane (SDN controller) to give forwarding instructions when new packet flows emerge.

2. Programmability of network services—The administrator may introduce complex rules and programs for the control layer, which are then consistently executed in the data plane.
3. Logically centralized control—network administrators can program the behaviour of the traffic in a centralized manner.

SDN is typically used with network function virtualization (NFV) technologies [28]. NFV provides a virtualization framework where it is possible to create mobile network entities and services as Virtual Network Functions (VNFs) on demand and place them at the most suitable location using the most appropriate amount of resources.

Network slicing [29] is a concept based on SDN. It enables the deployment of multiple logical networks independently on a common physical infrastructure platform. Network slices are created on-demand and they are isolated and restricted to the assigned resources. Micro-segmentation [30] is a concept originating from data centres, but its viability has been considered [31] for mobile networks as well. In a 5G network, a micro-segment can be defined as a network portion, which has been dedicated to a particular application or user and which protects particular (critical) network functions with the same security requirements. While an end-to-end slice contains all functions needed to create 5G connectivity, micro-segments may contain only one or a few functions that are secured using micro-segment specific means and policies. Micro-segments may be utilized to deploy fine-grained isolation, specific access controls, and tuned security policies and controls that fulfil application-specific trust models.

SDN provides inherent capabilities for monitoring as switches feed information to a controller that can analyse the data in a centralized manner and the network can easily react to data flows related to threats. However, SDN introduces new challenges. Liyanage et al. [32] presented attack vectors that the introduction of SDN will bring for mobile networks. New interfaces, control protocols, and centralized architecture may be vulnerable for attacks and reduced diversity enables attacks to propagate quickly.

Researchers have proposed several concepts and tools for monitoring communication flows in SDN. For instance, OpenNetMon by van Adrichem et al. [25] is an approach and open source tool for monitoring data flows. It focuses on Quality of Service (QoS)-related metrics, in particular, polling statistics from edge switches at an adaptive rate in order to verify that throughput, delay, and packet loss are acceptable. OpenTM by Tootoonchian et al. [26] utilized OpenFlow statistics for traffic estimation in order to, e.g., optimize load balancing in SDN or to detect anomalies. OpenSAFE by Ballard et al. [27] utilized OpenFlow-based SDN to route selected traffic to monitoring appliances for deeper analysis without impacting network performance. Donatini et al. [33,34] developed an SDN-based monitoring platform for 4G networks to detect issues causing complexity and high bandwidth. Youssaf et al. described technologies [35] These tools have mainly focused on quality and performance issues in general. However, they are also helpful when monitoring security attacks affecting quality and performance.

3. Framework for Security Monitoring and Multi-Domain Trust Management

This section presents a scalable and flexible framework for security monitoring, inferencing, autonomous protection, and trust knowledge sharing between different stakeholders. The framework builds on SDN, information distribution, and inferencing solutions. First, we present an architecture that illustrates essential components and their requirements. Then, we describe in more detail the opportunities that SDN brings for monitoring. Finally, we describe a mechanism for sharing security and trust information between domains.

3.1. The Framework

The monitoring framework, illustrated in Figure 3, is a collection of enablers and features (software components for extracting security awareness from 5G networks) and information sharing mechanisms for tying these enablers together. The framework consists of the following enablers:

1. The micro-segmentation enabler facilitates creation and control of slices. It organizes and isolates network traffic flows. The enabler is a software component that uses a virtualization platform, access control functions, and an SDN controller to create slices and manage and adapt traffic flows.
2. Monitoring brokers distribute security event information. Brokers combine event flows from parts of the network within the end-to-end slice and make them available for different security inferencing functions.
3. The inferencing platform and functions generate security awareness from monitored data. The platform provides libraries for correlating and analysing large amount of streaming data flows.
4. Security adaptation mechanisms change the behaviour of 5G networks and security control mechanisms based on inferred knowledge on risks and trust levels.
5. The Trust Level Agreement (TLA) mechanism and Trust Metric Enabler facilitate knowledge exchange across administrative domains. The TLA enables the orchestration of end-to-end trustworthy slices.

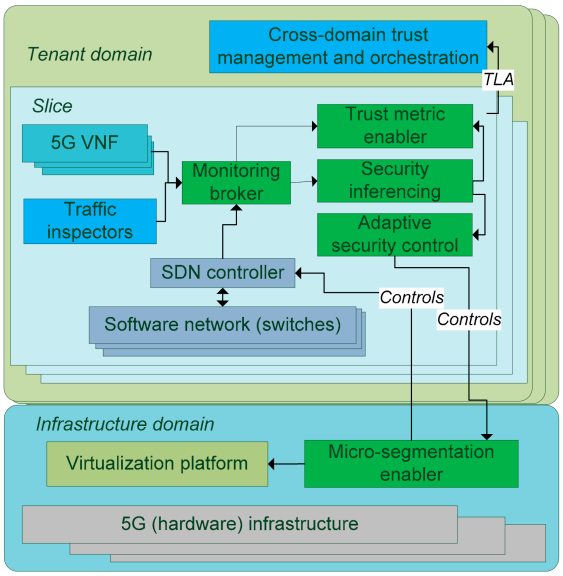

Figure 3. The monitoring framework. VNF: Virtual Network Function; SDN: Software-Defined Network; TLA: Trust Level Agreement.

3.2. Event Brokering

The glue in the monitoring framework is the information brokering mechanism. It enables different parties to connect each other and share information and knowledge. The proposed paradigm for information distribution is to 'publish and subscribe'. In this paradigm, the information is published to a central element broker, which then forwards information to components that have subscribed to that particular information flow. The approach increases the scalability and flexibility of 5G security monitoring as:

- It is easy to add new heterogeneous event sources to the system without changes to the broker or existing inferencing applications.
- Inferencing components can be integrated to the system for processing and inferencing of event streams as needed and reused in different applications.
- Events are efficiently provided only for those components that are interested on them.

Security event information sources include, for instance, SDN controllers that feed the broker with SDN topology, and authentication events, as well as traffic statistics. Different 5G network

functions (e.g., for mobility management, routing, authentication, traffic acceleration) may also collect and publish security event information while deep traffic inspectors intercept and analyse traffic flows—headers and payloads—in detail.

The framework deploys 'big data' technologies for information brokering and inferencing. Existing general purpose complex event processing as well as publish-and-subscribe platforms, supporting cluster computing, are utilized to handle large numbers of event streams and infer knowledge in near-real time.

3.3. Security Inferencing

The inferencing platform and its functions provide dynamic security responsiveness for the network. The platform functions realize the autonomous control loop [36] in which analyses monitored events, plans control actions, and then executes those actions. Analysis functions include rule-based threat detectors and anomaly detectors, which are based on machine learning. Inferencing functions are supported by different reusable complex event processing functions that merge, correlate, or aggregate event information flows. Rule-based reasoning determines the risk level for the anomaly in the network context (e.g., heavy traffic from a device maybe interpreted as potential DoS attack). The planning phase selects a mitigation function (e.g., quarantining a suspected node from the network). The execution phase then requests the mitigation function to perform a control action, for instance, for SDN to block a suspected node.

3.4. Customizing Security by Software-Defined Networking

The framework utilizes SDN to isolate traffic flows that are related to different applications or users. Consequently, different monitoring, inferencing, and control functions may be applied for different applications. The approach provides scalability to 5G monitoring in the sense that it allows to use more computationally expensive algorithms as different algorithms may be dedicated to different data flows. For instance, some IoT applications may be more vulnerable to availability threats and hence analysis on an IoT-specific slice may focus more on such threats and less on threats related to privacy or tracking of a terminal's location.

The framework includes a micro-segmentation enabler that is used to: (1) control and create virtualized network slices; (2) control access to SDN; (3) collect monitoring information from SDN; and (4) enable cooperation between SDNs in different domains. The enabler parses traffic statistics available from software switches and sends them in a format supported by the broker. The enabler can support different authentication mechanisms. The cooperation between different slices in single administrative domain is sustained by creating secure tunnels between them.

Deep traffic analysis is enabled by redirecting particular data flows to traffic inspectors. Different inspectors may be applied to examine headers and data payloads in order to detect protocol non-compliance, malicious content, or other policy violations. Packet inspectors need to be delivered decryption keys for encrypted data payloads that they are authorized to inspect.

3.5. Trust Level Agreement between Slices and Domains

SDNs in different domains provide different security levels. For instance, operators may have different security practices and security may be customized for applications. The users of 5G slices can be organizations, service providers, end-users, or cooperating operators. They all need to be continuously and strictly aware of how trustworthy the network services they get are, and whether the network fulfils their security requirements.

Security information shared between domains must be at least near-real-time and thus support dynamic scenarios where the client, which is, for example, an end-to-end multi-domain orchestrator, may at any time change to another operator or network. However, disclosing all security-related information is not often feasible due to the volume and complexity of information as well as the

sensitivity of the information for operators—it may reveal company secrets, privacy critical data, or weaknesses that could be utilized by attackers.

The Trust Level Agreement (TLA) mechanism enables sharing of near-real time measurements that can be used in end-to-end connectivity management and orchestration. The TLA mechanism enables 5G users to specify and track the required level of security for the network. Users can specify arbitrary trust models and requirements that must be fulfilled before they accept the network. The TLA is implemented by the Trust Metric Enabler. It ensures that trust requirements are continuously fulfilled and informs users of critical changes in the monitored trust levels. The enabler may also be used to hide domain-specific complexities and details that may be sensitive from a privacy or operator perspective. There may be explicit controls on what information is disclosed. Also, as the trust metrics are calculated per application-specific slice, the enabler inherently ensures that information on other slices and applications does not leak.

The messaging diagram, in Figure 4, illustrates four simplified phases of TLA communication. In the first phase, the Trust Metric Enabler monitors network for security relevant events and measurements. In the second phase, a client, wishing to use a 5G slice, requests the enabler to resolve whether the given slice fulfils the client's trust demands. In the third phase, the client has found a network slice it trusts. The service provider and the client agree the use of it and establish a secure tunnel for communication. The fourth phase illustrates how the Trust Metric Enabler continuously monitors the network and informs the client of critical status changes.

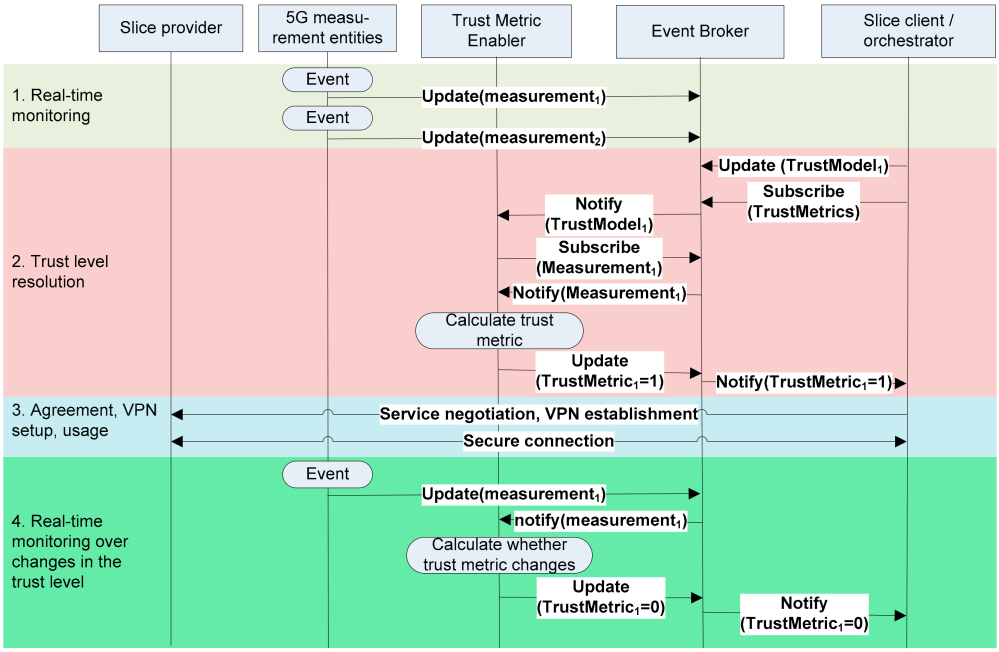

Figure 4. Trust Level Agreement messaging—The framework aggregates, filters, and brokers of security information between domains (updated from the Trust Metric Enabler's open specification [37]). VPN: Virtual Private Network.

4. Security Scenarios of Multi-Domain Data Correlation

This section presents two scenarios for 5G security. We will first describe the threats and then propose novel approaches based on multi-domain metrics for assessing the existence of threats.

4.1. IoT Authentication Storms and Adaptive Group Authentication

In 5G systems, new emerging IoT applications cause additional threats to availability [38,39]. The previous mobile network generations were designed for user devices that authenticated at an even

pace. In 5G systems, simultaneous authentication events may cause an excessive traffic and processing burden for 5G Authentication, Authorization and Accounting (AAA) services. Pre-5G authentication services (particularly Home Subscriber Server, HSS) were not designed to handle situations where thousands or millions of devices authenticated simultaneously. In IoT, devices often behave differently to phones by sending short periodic communication bursts, e.g., they may switch on or send reports at even hours or when the sun starts to shine. Thus, many devices may authenticate to the network at the same time. Furthermore, adversaries—aiming to perform a denial-of-service attack—may use different means to initiate traffic spikes. As a result, authentication processing may face a heavy overload situation and thus 5G service becomes unavailable. Solutions to the threat include blocking of detected adversaries [40] as well as gateways and group authentication schemes [41,42] where the serving network handles more of the authentication signalling load.

In this scenario, illustrated in Figure 5, we monitor signalling in three domains involved in authentication: access, serving, and home network domains. The collected monitoring data includes SDN traffic statistics towards authentication components as well as the used and available authentication mechanisms.

An overload situation in one domain may then trigger different control actions:

1. In a situation where a home network has enough capacity, the system allows devices to authenticate directly with the home network using, e.g., a standard authentication and key agreement (AKA) [5] protocol.
2. In a case where a monitor detects an overload situation in the home network domain, it adapts security controls in access and/or user equipment domains so that traffic flows are blocked and devices are required to use group authentication mechanisms instead. These mechanisms require less effort from the home network but may also have some downsides with respect to the access network performance or for the security level [43].
3. In a situation where the access network does not have enough capacity for group authentication, the monitoring system may allow device access through trusted gateways which authenticate devices locally. This solution is the most scalable but places trust in local gateways.
4. In a situation where a slice subscriber has specified a trust model with high integrity requirements, the strongest (e.g., direct authentication with home network) is applied even if that means that some devices will be disconnected. The slice may also have set particular trust models, which define the amount of devices that can be connected to a slice using group authentication mechanisms and for which authentication mechanisms are possible.

The controls can be implemented through the SDN means by redirecting traffic flows to a slice that provides authentication services suitable for each situation.

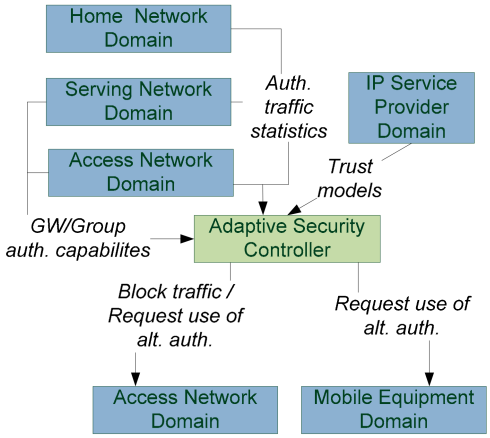

Figure 5. Domains and information in the authentication storm scenario. GW: gateway.

4.2. Location Tracking and Adaptive Privacy Protection

A device's location can be tracked by eavesdropping identifiers that are transmitted in clear text between a base station and a user terminal. For instance, the location can be tracked [44,45] by using Globally Unique Temporary Identifier (GUTI) or Temporary Mobile Subscriber Identity (TMSI). Devices use these identifiers to detect signals that are targeted for them and hence they are sent unencrypted. The identifiers are pseudonyms but are randomly changed (typically only in roaming situations). Broadcasting a temporary identifier, which is known or suspected to belong to Alice, is an indication that Alice is close to the broadcasting base station. Hence, if such identifiers are not changed (re-pseudonymized) before an adversary is able determine which identifier belongs to a victim, the victim's location can be tracked. In 5G systems, the cell sizes in urban environments are becoming smaller, and thus location tracking is becoming more accurate. In a typical attack, Mallory triggers communication (a harmless looking short message, email, internet call, or instant message) to Alice that will cause signalling in radio interfaces where Mallory is listening.

The solution to this location tracking threat is to change (re-pseudonymize) temporary identifiers often enough but not so often that excessive signalling is caused. The challenge in this approach is to know when to change identifiers.

We propose using a monitoring system (adaptive security controller in Figure 6) to track application layer communication in order to infer what the optimal re-pseudonymization time is. Identifiers do not need to be changed if there have not been any traffic flows that could had been triggered by an adversary. The security controller must have application domain knowledge on the potential sources of Mallory's reconnaissance messages. For instance, Mallory may trigger communication through Alice's email and social media servers. The monitor, in the access network domain or network domain, may then track the incoming messages (large encrypted packets) from these server addresses. In case, when the security controller sees such messages, it triggers re-pseudonymization in the access network domain.

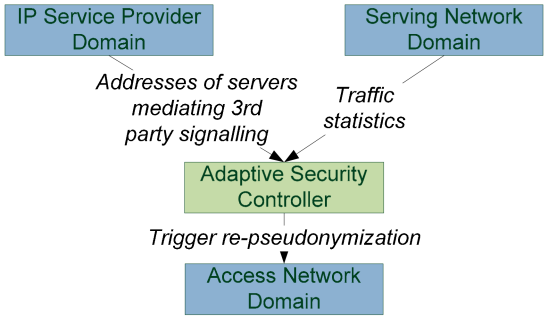

Figure 6. Domains and information in location tracking scenario.

The identifiers must be reallocated at the same time for all the nodes in the same cell and slice. If reallocation were to be done only for the targeted node, the mechanisms might leak information as the adversary, after initiating malicious flows, would notice a new identifier emerging.

The adaptive privacy protection maximizes privacy protection while trying to optimise reallocation signalling. The approach provides operators a chance to differentiate privacy services. As the approach requires more monitoring, it can be provided as an optional service that is available in a particular slice that has been created with SDN and dedicated for particular application. For many applications, such as for vehicles and medical devices, there may be less revealing connections and reallocations can be sparse but the privacy threat may be large. However, the solution may be unfeasible for some applications. For example, for some IoT equipment with fixed geostationary locations, tracking is not a threat and thus identifier reallocations are not needed. For some cells with lots of active devices with frequent messaging and social media applications, the signalling overhead due to mechanism can be large. In these cases, an operator may consider the most feasible reallocation

frequency, considering whether the operator should provide a service with a high privacy level where reallocations are executed after every potential compromising flow or whether the operator should delay reallocations.

The amount of worst case signalling overhead caused by adaptive privacy protection can be estimated. The worst-case pseudonyzation frequency is given by the number of revealing messages divided by the number of messages that is needed to map a pseudonym to users identity multiplied by the number of average connected devices to a base station. For instance, it may be assumed that the average number of social media messages is 32 [46], the number of users connected to a base station is 750 [47], and that an attacker needs to send 10 social media messages for mapping [45]. In this case, the worst case amount of reallocations is 2325 per day. However, in practise the revealing messages will arrive in overlapping time periods: after a reallocation is made an adversary must send the ten probing messages again.

5. Implementation

This section describes our implementation of the framework and enablers. Detailed interface specifications [12] and user manuals [48] are available from the 5G-ENSURE project website [49]. Figure 7 illustrates the components and technology selection for the prototype implementations. It also highlights the information distribution aspects of the framework. It shows how different elements are connected and how the security-related information flows. The architecture separates the network and infrastructure specific security event sources (below the broker) from the application- and resource-specific security processing and inferencing (above the broker). The figure illustrates planned event publishers (blue in the figure), event consumers (i.e., inference components—green in the figure), and event distribution and processing framework (red in the figure).

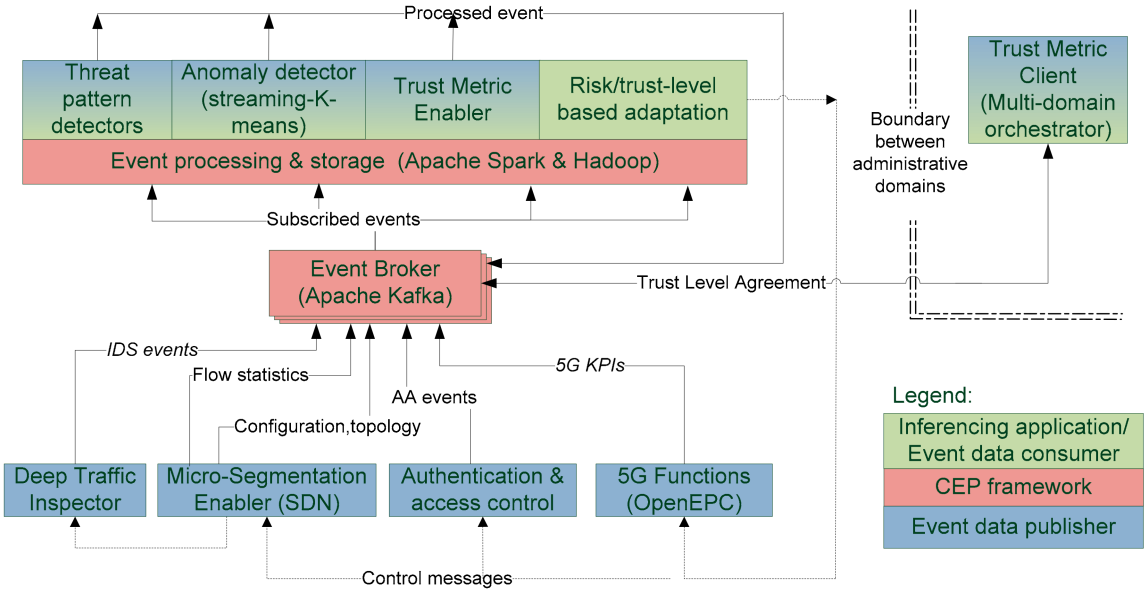

Figure 7. Framework for distribution of security event information—showing technology selections for prototype implementation (updated from the user manual of the monitoring enabler [48]). EPC: Evolved Packet Core; CEP: Complex Event Processing; AA: Authentication and Authorization; IDS: Intrusion Detection System.

5.1. Micro-Segmentation Enabler

A micro-segmentation enabler is a software component facilitating creation, deletion and control of slices within SDN. The enabler uses a modified OpenVirteX virtualization platform [50,51] to create slices and the Ryu SDN controller [52] to manage and adapt traffic flows. The enabler has been

integrated with different authentication protocols including the Extensible Authentication Protocol -Authentication and Key Agreement (EAP-AKA) [5] and the EAP Method for MD5 hash (EAP-MD5), which is a computationally light SIMless alternative for IoT devices. Furthermore, the enabler supports multi-domain cooperation by supporting the creation of IPsec VPN tunnels between different slices.

The development and testing was done on a Linux host on top of Mininet environment [53], which emulates OpenFlow-supported network and hosts, using Open vSwitch virtual switches [54] and Linux namespaces.

The micro-segmentation enabler provides an interface that can be used to publish traffic statistics as well as topology and authentication events to the monitoring enabler. Furthermore, it provides an interface that monitoring enablers can use to request control actions, especially to quarantine particular nodes from the network.

5.2. Sharing of Monitoring Data

The monitoring enabler utilizes Apache Kafka [55] as an event broker to distribute information between event producers and event subscribers. Kafka is a publish-and-subscribe system that has been designed to be fast, scalable, and durable. Kafka brokers can be clustered to provide more resources elastically and transparently. A broker keeps messages on disk and replicates them within a cluster to prevent data losses. Each broker should be able to store terabytes of messages and handle megabytes of reads and writes per second from thousands of clients. Event information from a broker in one domain may be subscribed and delivered to other brokers in order to further distribute decision making (to enable further scalability) and to enable domains to share information and detect cross-domain attacks, for example. To control information sharing across domains, Kafka provides authentication and authorization functions.

The framework is targeted for near-real-time processing of monitored data streams. However, in security inference, history data is also needed, e.g., short-term history for correlation analysis, and a longer history for machine learning. Smaller amounts of history data can be stored by the broker and for longer term history data the inference components must be integrated with a database for storing relevant events. Kafka integrates easily with Hadoop, which is the recommended database for the framework deployments.

5.3. Security Inferencing and Anomaly Detection

The platform for security inferencing is Apache Spark [56]. Spark is a general engine for cluster-based data processing originating from UC Berkeley. It provides specialized data processing libraries (including Structured Query Language (SQL) and DataFrames, MLlib for machine learning, GraphX, and Spark Streaming), which may be combined to create parallelized applications. Spark is designed to be fast, and by supporting mainstream languages (Java, Scala, Python and R languages) it is easily accessible for developers. It can be run as a stand-alone mode or, e.g., within Hadoop—which is a framework for distributed storage (using Hadoop Distributed File System, HDFS) and processing (using MapReduce) of large sets of (history) data.

The *Monitoring Enabler* is a Python application that subscribes and processes event information in order to, for example, detect anomalies or occurring attack patterns or to generate security status information for other security/trust components. Inferencing information users at the end of the chain, the *Trust Metric Enabler* and *Adaptive Security Controller* are then expected to infer knowledge on slice's security situation, i.e., provide trust metrics or initiate some security responses.

The anomaly detection mechanism integrated into the current implementation is streaming-*k*-means clustering [57,58] which is an unsupervised machine-learning algorithm. Streaming-*k*-means is an adaptation of a K-means algorithm [59,60], which partitions a set of data points (feature vectors) into *k* number of clusters. The original *k*-means algorithm learns a clustering model during a training period. Anomalies are then found by calculating the distance of new data points from the cluster centre points and determining whether the distance is larger than the predefined threshold. Streaming-*k*-means

provides support for forgetfulness; the algorithm can learn over time how the clustered data changes. The algorithm is therefore suitable for dynamic environments where devices are connecting and disconnecting to the network and initiating new connections (even in normal non-attack situations). A challenge for the algorithm is to detect attacks that are gradually strengthening or which are variations of previously seen attacks (if those have not been excluded from the clustering model).

5.4. Trust Metric Enabler

The Trust Metric Enabler is a Python application that implements the Trust Level Agreement Mechanism, specified in Section 3.5. It simplifies the security monitoring from the client's perspective. The enabler is requested to follow particular 5G measurements and events and determine how well they match a trust model specified by the client. The enabler notifies clients only if changes occur in the network that affect client's trust towards it. The implementation utilizes a Kafka framework to collect measurements as well as to exchange trust models and trust metric information with clients at run-time.

Trust models specify requirements that the 5G network must fulfil to provide the required trust level. All requirements stated in the model must be fulfilled to confirm that the network is trusted. The current implementation supports three types of measurements:

- *Service*—Stating that a particular service is running. "The service must be running for the network to be trusted".
- *Max_level*—Setting an upper boundary for particular aggregated events. "There must be at most x number of events y occurrences for the network to be trusted."
- *Min_level*—Setting a lower boundary for particular aggregated events. "There must be at least x number of events y occurrences for the network to be trusted."

Trust models are presented using JavaScript Object Notation (JSON) files. A simple example is given in Figure 8. The example contains an identifier of the model and a requirement that two particular enablers must be running, as well as requirements that the number of devices that have been authenticated using MD5-based AKA is at most 50, and that the anomaly level is 1.

```
{ 'policyid': 1234,
  'services':
   {    'msidx.securitymonitoringenabler': 1,
        'msidx.microsegmentationenabler': 1   },
  'maxlevels':
   {    'msidx.authcounter.MD5': 50,
        'msidx.anomality_level': 1   }
}
```

Figure 8. A trust model example.

Trust models are sent to the Trust Metric Enabler, which provides metrics as a reply, first when the request is made and then again if the metric changes. Metrics are presented as a simple JSON file, with the model's identifier (as specified by the Trust Model Creator) and its value. The enabler counts the amount of rules in each trust model and can be used to filter models that provide only few rules and hence may, for example, leak privacy critical information.

5.5. Mobile Network Testbed

To evaluate the framework, we developed a testbed composed of example applications (IoT and video streaming) and software implementations of mobile network's core functions, as well as

emulated User Equipment (UE) and radio access network functions. The testbed produces realistic traffic patterns that were then analysed using machine learning algorithms (See Section 6).

The testbed's devices comprised (see Figure 9) one laptop computer connected via WiFi and Ethernet to a server in the local laboratory network named Willab. User Equipment (UE), Evolved Node B (eNB), and all Evolved Packet Core (EPC) components were run on the laptop. For those, we used OpenEPC Rel. 6 [61] that implements the EPC components and emulates eNB and UE functionalities in various virtual machines. In our case, the UE, eNB, and Packet Data Gateway (PGW) each had their own virtual machine. The Serving Gateway (SGW) and EPC-Enabler virtual machines contained more than one component, e.g., the SGW's virtual machine supported the Mobility Management Entity (MME) in addition to the SGW. The EPC-Enablers also contained the Media Delivery Function (MDF) server that was used for streaming video to the UE. The virtual machines were connected to each other through virtual network interfaces (net_a/b/c/d in the figure). In our setup, the UE used an Long Term Evolution (LTE) connection to the eNB and the EPC. The OpenEPC also supports WiFi connections, but we did not utilize them in these tests. Traffic can be generated by activating any application in the UE or on the server. We implemented the libcoap [62] client and server to generate IoT traffic that uses the Constrained Application Protocol (CoAP) [63]. The Session Initiation Protocol (SIP) [64] was needed to initiate and control the video streaming sessions over a Real-time Transport Protocol (RTP) [65]. The traffic data from each interface was captured with Wireshark [66]. The captured packet traces can then be further analysed with security inferencing algorithms from Spark.

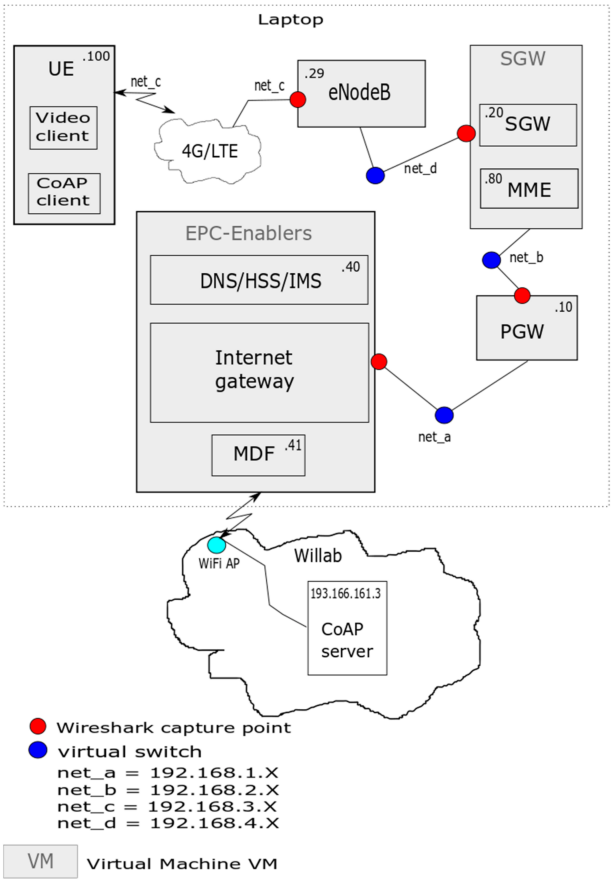

Figure 9. The testbed. UE: User Equipment; SGW: Serving Gateway; MME: Mobility Management Entity; CoAP: Constrained Application Protocol; PGW: Packet Data Gateway; DNS: Domain Name System; HSS: Home Subscriber Server; MDF: Media Delivery Function; LTE: Long Term Evolution; IMS: Internet Protocol Multimedia Subsystem.

6. Characteristics of Application-Specific Data in the Mobile Network Testbed

We illustrated and studied how application isolation affects the characteristics of data flows in mobile networks by building two experimental arrangements with our mobile network testbed. Essentially, we studied the characteristics in a single application scenario—with homogeneous traffic flows—and then compared it to a multi-application scenario—with more heterogeneous flows. In the first scenario, we monitored a mobile network running an IoT application. In the second scenario, we analysed the mobile network supporting both video streaming and IoT applications.

6.1. EPC Data from Video and IoT scenarios

Data was collected from different measuring points in our testbed (as depicted in Figure 9). The measured time was roughly two minutes. The feature vector used in the analysis contained four individual packets: the destination and source IP address and timestamp, as well as used protocols. Most of the traffic consisted of IoT or video traffic over CoAP or RTP protocols, respectively. In the IoT only scenario, the number of CoAP packets comprised 53–57% at all measurement points except at net_d, where CoAP packets presented only 36% of all packets. The Address Resolution Protocol (ARP) covered also 36% at that point. In the video and IoT scenario, video produced 96–97% of all packets at all measurement points, while only 1% were from CoAP. This is due to the much greater CoAP packet interval of two seconds, which also explains the rather low packet percentages for CoAP in the IoT-only scenario. In the video scenario, also some SIP packets that control the video stream were detected. Otherwise, the rest of the packets belonged to the Spanning Tree Protocol (STP), the Internet Control Message Protocol (ICMP, ICMPv6), the Stream Control Transmission Protocol (SCTP), the Address Resolution Protocol (ARP), and the Domain Name System (DNS).

The heterogeneity of the data collected from the testbed is visualized in Figure 10 (data seen when only IoT applications were used) and Figure 11 (data seen in combined video and IoT case). The illustrations were done using a Multidimension Scaling (MDS) algorithm [67,68] that places our multi-dimensional data objects into a four-dimensional diagram (where colour is the fourth dimension) in a manner that the distances between objects are preserved as well as possible. The MDS algorithm provides a visual mean—by comparing the shapes of result diagrams—to measure the level of similarity of datasets.

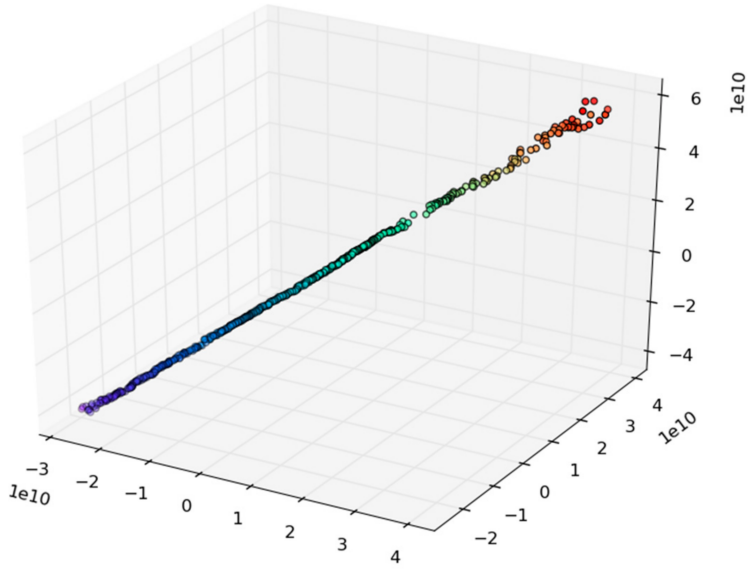

Figure 10. Multidimensional scaling illustration of the testbed communication with Internet of things (IoT) applications only.

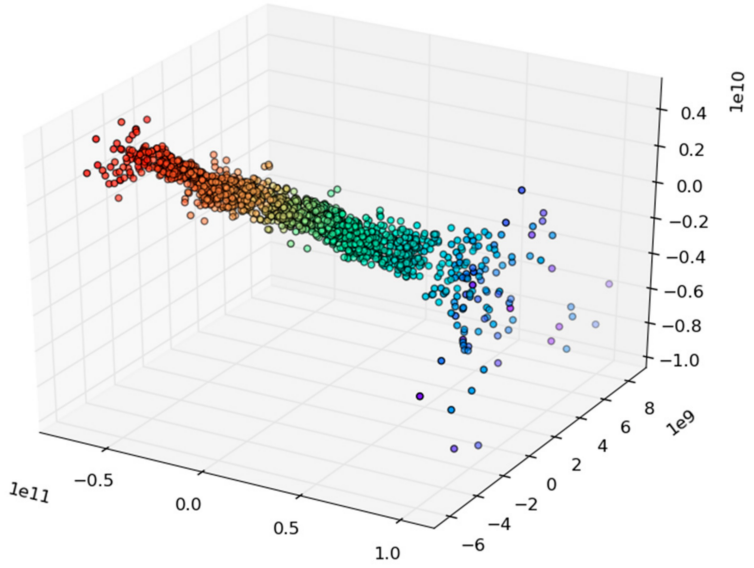

Figure 11. Multidimensional scaling illustration of the testbed communication with video and IoT applications.

The figures illustrate the increased diversity between the two scenarios. In practice, the difference comes from the fact that in the video and IoT scenario there are more packets, more protocols, as well as more data flows going from different destinations to different sources.

6.2. Effects on Machine Learning

The data sets were also analysed using machine learning algorithms in order to study how the homogeneity affects to data analysis. The used algorithms were *k*-means [59,60], which is a well-known clustering algorithm often used in anomaly detection, and bisecting-*k*-means [69], which is a fast variation of *k*-means where clusters are found hierarchically. We studied differences in the complexity of optimal models (i.e., the optimal cluster amount *k*). For the analysis, the collected data was first normalized using scripts based on scikit-learn library [70,71] and then analysed with of Apache Spark's *clustering* algorithms [72]. The results are only suggestive. The results of the *k*-means algorithm depend on the data set and may in some cases provide counterintuitive results. For this analysis, we used our IoT and video-IoT data sets, that have lots of similarities (the same core network signalling).

The *k*-means algorithm clusters data points into *k* amount of clusters. The accuracy of a learned cluster model is evaluated by measuring the variance (so called Within Set Sum of Squared Errors—WSSSE), which is the combined Euclidean distance of all data points to their nearest cluster centres. Figure 12 illustrates these distances in the two distinct data sets. The *y*-axis shows distance values, while the *x*-axis depicts the *k* value. In *k*-means the average error distances decreases while cluster amount value increases but on the other hand a large *k* value makes models less feasible. The effective *k* value locates in 'elbow' points of diagrams where a decrease of WSSSE slows down.

By looking the figures, we can see that in both cases, the distance to closest cluster centroids is smaller for the IoT data than it is for the combined IoT and video data (*y*-axis). However, the curves in the diagram cannot be directly compared as the scales for different data sets are different. Therefore, to find the optimal *k*-amount we compared diagrams using the shape of the diagram and by locating the 'elbow' points.

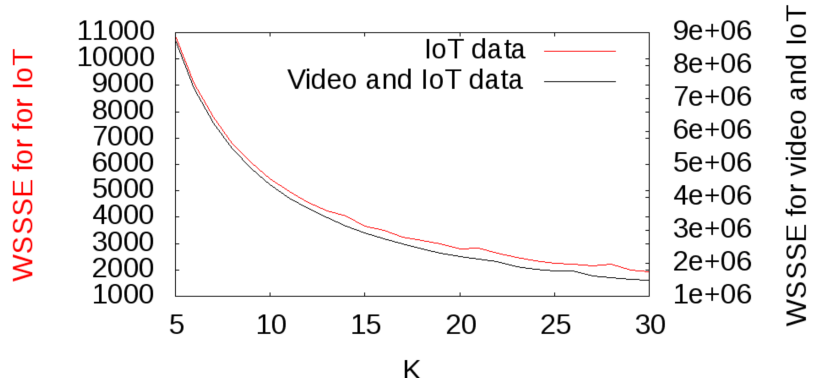

Figure 12. The *k*-means clustering for IoT only and video/IoT scenarios. The diagram illustrates average distances to cluster centroids with different numbers (*k*) of clusters. The left side y-columns provide values for IoT data (red curves) and the right side y-columns provide values for video/IoT data (black curves). WSSSE: Within Set Sum of Squared Errors.

When considering the complexity of the model (amount of clusters), the results from the *k*-means provide some indications that the same accuracy can be achieved with a smaller cluster amount in the IoT only case) than in the combined scenario. The *k*-means figure illustrates that the IoT-only curve turns upwards to a greater degree than the combined curve, so the elbow for the IoT only curve is more on the left. To obtain some numerical estimates about the differences in optimal *k* values, we looked at the ratio between the decrease in WSSSE value and the previous value. The decrease is less than 5% when *k* is 17 for the IoT-only data and 20 for the combined data. Hence, in this case, *k* is three clusters (15%) smaller for homogeneous data than for more heterogeneous flows. The fewer centroids there are, the more likely it is that random and uniformly-distributed 'malicious' data points generate a longer distance to the centroids and, hence will be correctly interpreted as malicious. Detection of anomalies is likely to be easier when the average distances are small as in these cases the detection threshold can be set lower and will catch more anomalous datapoints.

The results from bisecting-*k*-means (Figure 13) did not produce the same advantages in the optimal cluster count. The algorithm utilizes the hierarchical algorithm where clusters are split recursively as the algorithm progresses. The 'elbow points' were at the same locations for both the IoT and IoT/video data sets.

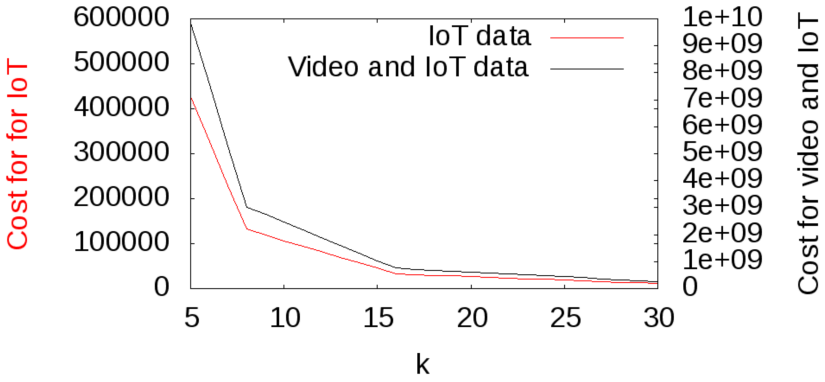

Figure 13. The bisecting-*k*-means clustering for IoT only and video/IoT scenarios. The diagram illustrates average distances to the cluster centroid with different numbers (*k*) of clusters. The left side y-columns provide value for IoT data (red curves) and the right side y-columns provide values for video/IoT data (black curves).

7. Discussion

Security monitoring and sharing of the security knowledge are essential enablers for ensuring the security of the 5G systems and the new emerging applications. Firstly, they support design time security planning as they enable different actors to develop and deploy mechanisms that address the security issues that emerge frequently. Secondly, they also support operational (run-time) protection of systems, as they enable different parties to exchange information on their security capabilities and readiness to fulfil other parties' requirements. They also advance run-time responsiveness to detected security incidents and automation of security protection.

Monitoring enables SDN orchestration, i.e., autonomous adjustment of network's behaviour and cooperation. Based on the knowledge of threats and the trust situation gained by monitoring and inferring, the security behaviour of a 5G network can be adapted in different manners. For instance, traffic flows can be filtered or redirected, nodes can be blocked from the network, alternative security protocols can be deployed, and security monitoring can be intensified.

SDN-based slicing enables security monitoring to focus on particular threats and traffic types. Algorithms that analyse the same information and detect the same threats from large heterogeneous traffic masses are possible—but more complex—to develop. It is simpler and easier to implement monitoring algorithms for homogeneous data streams. The development of modular monitoring solutions is more flexible, faster, and scalable as we can focus on the relevant threats only.

In principle, conventional network management, firewalling, routing, and monitoring solutions provide a means of separating applications and to focus monitoring on particular data flows. However, conventional approaches suffer from complexity as policies must be defined on top of large and versatile mobile network settings. With SDN, we can do slicing more efficiently and with fewer errors.

The proposed framework can detect different kinds of threats. In Section 4, we discussed location tracking and IoT botnet scenarios in more detail. Other potential use cases could include:

- Detecting man-in-the-middle (MitM) attacks by following times that it takes for specific packets to cross different domains and detecting if there are delays that could indicate MitM. Such an approach can be used to improve application domain security—e.g., to detect attacks against two-channel authentication/verifications used for example in bank applications or in company intranet authentication.
- Downgrading attacks—detecting cases where a capable device is forced to use connectivity alternatives (5G $\geq$ 3G, 4G, or WiFi) that may have weaker protection. Such attacks can be detected by monitoring the use of weak connectivity alternatives in locations where stronger mechanisms are available.
- Location spoofing by end-points—A device may want to spoof its location to an end-service in order to circumvent location-specific access controls. For instance, a sensor may be stolen and transferred to another location in order to spoof a data collector. Such attacks can be detected by monitoring which base stations are used to connect to the network. SDNs can also control access so that access to slices is only possible from particular locations.

Our SDN-based approach focuses on threats within network slices dedicated for particular applications. However, in many applications the adversary has access to several slices. For instance, the adversary may use one slice for control or reconnaissance purposes while another slice is used in actual attacks. An example environment for these cross-slice threats could be, for instance, a connected car that needs access both to "navigation" and "entertainment" slices. When a car is detected to be involved in an attack within a "navigation" slice (say providing bogus traffic statistics), the domain might drop this car also from the "entertainment" slice (as this slice may be used as a control channel for the navigation attack).

This means that the solutions monitoring different slices should share and correlate information. Consequently, some of the advantages brought by focusing only on homogeneous data streams are lost. However, the amount of information that must be shared across slices may still be insufficient

with respect to the amount of information that the monitoring of one heterogeneous unsliced network would require. At the maximum, information would need to be shared only between those slices that have common users.

8. Conclusions and Future Work

We studied how to utilize software networking and virtualization concepts to slice and segment a 5G network into small portions that are easy to manage and monitor. We also proposed an SDN-based monitoring framework that separates different phases of monitoring: information collection, distribution, inferencing, and control actions. The framework provides flexibility and eases application-specific and spatial customizations of security monitoring and control solutions for 5G. Also, to enable more accurate threat detection, our framework supports cross domain (between different 5G stakeholders) exchange of security knowledge and trust measurements. We evaluated the feasibility of our proposals by describing two monitoring scenarios—with cross domain-correlated metrics and with novel control proposals.

We also argued that monitoring accuracy increases as customized monitoring can focus on few specific data flows. The results we gained from the testbed illustrate that SDNs dedicated to only one application provide more homogenous data flows than SDNs that support more applications. This indicates that application-dedicated software networks can simplify monitoring and autonomous control and thus decrease the numbers of false positives and false negatives in anomaly detection. The initial results on monitoring accuracy should be verified with larger data sets and with different applications. We utilized MDS, *k*-means, and bisecting-*k*-means algorithms for data analysis. In the future, more work will be needed to evaluate how different learning models, such as *k*-means||, spectral clustering, Density-based spatial clustering of applications with noise (DBSCAN), may be applied and which models are suitable for which applications. More work also is needed to understand how well these approaches respond to different kinds of attacks. Specific attention should be given to cross-slice threats where an adversary has access to many applications and slices.

Acknowledgments: This work has been performed within the 5G-ENSURE project (www.5gensure.eu) and funding has been received from the European Union's Horizon 2020 research and innovation programme under grant agreement number 671562. The work has also been supported by the CORNET project (5gtnf.fi/projects/cornet/), partly funded by Business Finland. The authors wish to thank Arto Juhola for advice and tools related to machine learning, data normalization, and data visualization, as well as Juha Pärssinen for advice on big data frameworks.

Author Contributions: Jani Suomalainen designed and implemented security monitoring and trust metric enablers. He specified the use cases and performed data analysis. He was also the main contributor for the introduction, related work, discussion and conclusions sections. Kimmo Ahola provided the micro-segmentation enabler for softwarization and virtualization. Mikko Majanen provided the mobile network testbed and performed the experiments. Olli Mämmelä supported in specifying the softwarization and virtualization concepts and surveyed related work on micro-segmentations. Pekka Ruuska participated to the design of monitoring and trust metric enablers as well as supported the writing process.

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

References

1. ITU. *ITU towards IMT for 2020 and Beyond—IMT-2020 Standards for 5G*. 2017. Available online: http://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/imt-2020/Pages/default.aspx (accessed on 28 September 2017).
2. 3GPP. TR 38.913 Study on Scenarios and Requirements for Next Generation Access Technologies. Technical Report, v14.3. 2017. Available online: http://www.3gpp.org/ftp/Specs/archive/38_series/38.913/38913-e30.zip (accessed on 13 November 2017).
3. Gupta, A.; Jha, R. A Survey of 5G Network: Architecture and Emerging Technologies. *IEEE Access* **2015**, *3*, 1206–1232. Available online: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7169508 (accessed on 10 August 2015). [CrossRef]
4. 3GPP. *Technical Specification Group Services and System Aspects; 3G Security*; Security Architecture. (TS 33.102); 2012.
5. GPP. *TS 33.401 V12.14.0 (2015-03). 3GPP System Architecture Evolution (SAE). Security Architecture.* 2015. Available online: http://www.3gpp.org/ftp/specs/html-info/33401.htm (accessed on 28 September 2017).

6. 3GPP. TS 35.215. Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2. Document 1: UEA2 and UIA2 Specifications; Available online: http://www.3gpp.org/DynaReport/35215.htm (accessed on 28 September 2017).
7. 3GPP. 3G Security; Access Security for IP-Based Services. 3GPP Specification. TS 33.203 V12.6.0. 2014. Available online: http://www.3gpp.org/DynaReport/33203.htm (accessed on 28 September 2017).
8. 3GPP. *Technical Specification Group Services and System Aspects; 3G Security; Network Domain Security (NDS)*; IP Network Layer Security (TS33.210); 2015.
9. Traynor, P.; Lin, M.; Ongtang, M.; Rao, V.; Jaeger, T. On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core. In Proceedings of the 16th ACM conference on Computer and Communications Security, Chicago, IL, USA, 9–13 November 2009; Available online: http://dl.acm.org/citation.cfm?id=1653690 (accessed on 29 May 2017).
10. Mulliner, C.; Seifert, J.-P. Rise of the iBots: Owning a Telco Network. In Proceedings of the 2010 5th International Conference on Malicious and Unwanted Software, Nancy, France, 19–20 October 2010; pp. 71–80. Available online: http://ieeexplore.ieee.org/document/5665790/ (accessed on 29 May 2017).
11. G-ENSURE Project. *Security Architecture (Draft)—Deliverable D2.4.* 2016. Available online: https://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.4-SecurityArchitectureDraft.pdf (accessed on 17 August 2017).
12. 5G-ENSURE Project. 5G-PPP Security Enablers Open Specifications (v2.0)—Deliverable D3.6. 2017. Available online: https://5gensure.eu/sites/default/files/5G-ENSURE_D3.65G-PPPsecurityenablersopenspecifications%28v2.0%29.pdf (accessed on 17 August 2017).
13. 3GPP. Key Performance Indicators (KPI) for Evolved Universal Terrestrial Radio Access Network (E-UTRAN): Definitions (Release 8). TS32-450; 2011; Available online: www.3gpp.org/DynaReport/32450.htm.
14. 3GPP. Telecommunication Management; Key Performance Indicators (KPI) for Evolved Universal Terrestrial Radio Access Network (E-UTRAN): Requirements. TS 32.451; 2008; Available online: http://www.qtc.jp/3GPP/Specs/32451-a00.pdf (accessed on 6 March 2017).
15. ETSI. *Network Functions Virtualisation (NFV) Release 3; Security; Security Management and Monitoring; Draft*; GS NFV-SEC 013 V0.0.9.; ETSI: Sophia Antipolis, France, 2017.
16. ETSI. *NFV Security; Security and Trust Guidance; GS NFV-SEC 003—V1.1.1.* 2014. Available online: http://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/003/01.01.01_60/gs_NFV-SEC003v010101p.pdf (accessed on 10 February 2017).
17. Martin, A.B.; Marinos, L.; Rekleitis, E. Threat Landscape and Good Practice Guide for Software Defined Networks/5G. *ENISA*. 2015. Available online: http://openaccess.city.ac.uk/id/eprint/15504 (accessed on 14 August 2017).
18. Yan, Z.; Zhang, P.; Vasilakos, A.V. A security and Trust Framework for Virtualized Networks and Software-Defined Networking. *Secur. Commun. Netw.* **2015**, *9*, 3059–3069. Available online: http://doi.wiley.com/10.1002/sec.1243 (accessed on 6 June 2017). [CrossRef]
19. Luo, S.; Dong, M.; Ota, K.; Wu, J.; Li, J. A Security Assessment Mechanism for Software-Defined Networking-Based Mobile Networks. *Sensors* **2015**, *15*, 31843–31858. Available online: http://www.mdpi.com/1424-8220/15/12/29887 (accessed on 16 August 2017). [CrossRef] [PubMed]
20. Ahmed, M.; Naser Mahmood, A.; Hu, J. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **2016**, *60*, 19–31. Available online: http://www.sciencedirect.com/science/article/pii/S1084804515002891 (accessed on 6 September 2017). [CrossRef]
21. Gupta, A.; Verma, T.; Bali, S.; Kaul, S. Detecting MS Initiated Signaling DDoS Attacks in 3G/4G Wireless Networks. In Proceedings of the 2013 Fifth International Conference on Communication Systems and Networks (COMSNETS), Bangalore, India, 9–10 January 2013; pp. 1–60. Available online: http://ieeexplore.ieee.org/document/6465568/ (accessed on 29 May 2017).
22. Shabtai, A.; Tenenboim-Chekina, L.; Mimran, D.; Rokach, L.; Shapira, B.; Elovici, Y. Mobile malware detection through analysis of deviations in application network behavior. *Comput. Secur.* **2014**, *43*, 1–18. Available online: http://www.sciencedirect.com/science/article/pii/S0167404814000285 (accessed on 6 June 2017). [CrossRef]
23. Marquezan, C.; Mahmood, K.; Zafeiropoulos, A. Context Awareness in Next Generation of Mobile Core Networks. *arXiv*. 2016. Available online: https://arxiv.org/abs/1611.05353 (accessed on 14 August 2017).
24. López, L.B.; Caraguay, Á.V.; Vidal, J.M.; Monge, M.A. S.; Villalba, L.J.G. Towards Incidence Management in 5G Based on Situational Awareness. *Future Internet* **2017**, *9*, 3. Available online: http://www.mdpi.com/1999-5903/9/1/3/htm (accessed on 14 August 2017). [CrossRef]

25. Van Adrichem, N.; Doerr, C. OpenNetMon: Network monitoring in OpenFlow Software-Defined Networks. In Proceedings of the 2014 IEEE Network Operations and Management Symposium (NOMS), Krakow, Poland, 5–9 May 2014; Available online: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6838228 (accessed on 13 June 2016).
26. Tootoonchian, A.; Ghobadi, M.; Ganjali, Y. *OpenTM: Traffic Matrix Estimator for OpenFlow Networks*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 201–210. Available online: http://link.springer.com/10.1007/978-3-642-12334-4_21 (accessed on 16 June 2016).
27. Ballard, J.R.; Rae, I.; Akella, A. Extensible and scalable network monitoring using OpenSAFE. In Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking, San Jose, CA, USA, 28–30 April 2010; p. 8. Available online: http://dl.acm.org/citation.cfm?id=1863141 (accessed on 16 June 2016).
28. ETSI. *GS NFV-SWA 001—V1.1.1—Network Functions Virtualisation (NFV); Virtual Network Functions Architecture.* 2014. Available online: http://www.etsi.org/deliver/etsi_gs/NFV-SWA/001_099/001/01.01.01_60/gs_NFV-SWA001v010101p.pdf (accessed on 27 October 2017).
29. NGMN Alliance. *Description of Network Slicing Concept. Report.* 2016. Available online: https://www.ngmn.org/uploads/media/160113_Network_Slicing_v1_0.pdf (accessed on 21 September 2017).
30. VMware. *Data Center Micro-Segmentation A Software Defined Data Center Approach for a "Zero Trust" Security Strategy. White Paper.* 2014. Available online: https://blogs.vmware.com/networkvirtualization/files/2014/06/VMware-SDDC-Micro-Segmentation-White-Paper.pdf (accessed on 21 September 2017).
31. Mämmelä, O.; Hiltunen, J.; Suomalainen, J.; Ahola, K.; Mannersalo, P.; Vehkaperä, J. Towards Micro-Segmentation in 5G Network Security. *European Conference on Networks and Communications (EuCNC 2016) Workshop on Network Management, Quality of Service and Security for 5G Networks.* 2016. Available online: https://www.researchgate.net/profile/Olli_Maemmelae/publication/310447736_Towards_Micro-Segmentation_in_5G_Network_Security/links/582d678b08aef19cb811738b.pdf (accessed on 31 January 2017).
32. Liyanage, M.; Abro, A. Opportunities and Challenges of Software-Defined Mobile Networks in Network Security Perspective. *IEEE Secur. Priv.* **2016**, *14*, 34–44. Available online: https://www.researchgate.net/profile/Madhusanka_Liyanage/publication/282648199_Opportunities_and_Challenges_of_Software-Defined_Mobile_Networks_in_Network_Security_Perspective/links/561bb10a08ae78721fa100f2.pdf (accessed on 16 February 2016). [CrossRef]
33. Donatini, L.; Garroppo, R.G.; Giordano, S.; Procissi, G.; Roma, S.; Foddis, G.; Topazzi, S. Advances in LTE network monitoring: A step towards an SDN solution. In Proceedings of the Mediterranean Electrotechnical Conference (MELECON), Beirut, Lebanon, 13–16 April 2014; pp. 339–343. Available online: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6820557 (accessed on 20 June 2016).
34. Adami, D.; Donatini, L.; Foddis, G.; Giordano, S.; Roma, S.; Topazzi, S. Design and development of management functions for distributed monitoring based on SDN-based network. In Proceedings of the Euro Med Telco Conference (EMTC 2014), Naples, Italy, 12–15 November 2014; pp. 1–5. Available online: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6996655 (accessed on 20 June 2016).
35. Yousaf, F.; Gramaglia, M.; Friderikos, V.; Gajic, B.; von Hugo, D.; Sayadi, B.; Sciancalepore, V.; Crippa, M.R. Network slicing with flexible mobility and QoS/QoE support for 5G Networks. In Proceedings of the 2017 IEEE International Conference on 4th Communications Workshops (ICC Workshops), Paris, France, 21–25 May 2017; Available online: http://ieeexplore.ieee.org/abstract/document/7962821/ (accessed on 23 January 2018).
36. Kephart, J.O.; Chess, D.M. The vision of autonomic computing. *Computer* **2003**, *36*, 41–50. Available online: http://ieeexplore.ieee.org/document/1160055/ (accessed on 26 September 2017). [CrossRef]
37. Hiltunen, J.; Ruuska, P.; Suomalainen, J. Trust Metric Enabler Open Specifications. *5G-ENSURE Project. Deliverable D3.6. 5G-PPP Security Enablers Open Specifications (v2.0).* 2017. Available online: https://5gensure.eu/sites/default/files/5G-ENSURE_D3.65G-PPPsecurityenablersopenspecifications%28v2.0%29.pdf (accessed on 28 September 2017).
38. Jover, R. Security Attacks against the Availability of LTE Mobility Networks: Overview and Research Directions. In Proceedings of the 2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC), Atlantic City, NJ, USA, 24–27 June 2013; Available online: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6618585 (accessed on 28 June 2016).

39. Jover, R.P. Security and impact of the IoT on LTE mobile networks. In *Security and Privacy in the Internet of Things (IoT): Models, Algorithms, and Implementations*; Taylor & Francis LLC, CRC Press: New York, NY, USA, 2016; Available online: http://www.ee.columbia.edu/~roger/LTE_IoT.pdf (accessed on 7 March 2016).
40. 3GPP. Study on Machine-Type Communications (MTC) and Other Mobile Data Applications Communications Enhancements. TR 23.887. 2013. Available online: http://www.3gpp.org/ftp/Specs/archive/23_series/23.887/23887-c00.zip (accessed on 22 September 2017).
41. Cao, J.; Ma, M.; Li, H. A group-based authentication and key agreement for MTC in LTE networks. In Proceedings of the 2012 IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, USA, 3–7 December 2012; pp. 1017–1022. Available online: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6503246 (accessed on 25 May 2016).
42. Lai, C.; Li, H.; Lu, R.; Shen, X. SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks. *Comput. Netw.* **2013**, *57*, 3492–3510. Available online: http://www.sciencedirect.com/science/article/pii/S1389128613002570 (accessed on 17 May 2016). [CrossRef]
43. Giustolisi, R.; Gerhmann, C. Threats to 5G Group-Based Authentication. In Proceedings of the 13th International Conference on Security and Cryptography (SECRYPT 2016), Lisbon, Portugal, 26–28 July 2016; SciTePress: Setubal, Portugal, 2016. Available online: http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1062093&dswid=6179 (accessed on 28 September 2017).
44. Kune, D.F.; Koelndorfer, J.; Hopper, N.; Kim, Y. Location leaks on the GSM air interface. In Proceedings of the Network & Distributed System Security Symposium (NDSS), San Diego, CA, USA, 5–8 February 2012.
45. Shaik, A.; Borgaonkar, R.; Asokan, N.; Niemi, V. Practical attacks against privacy and availability in 4G/LTE mobile communication systems. In Proceedings of the 23rd Annual Network and Distributed System Security Symposium (NDSS 2016), San Diego, CA, USA, 21–24 February 2016; Available online: https://arxiv.org/abs/1510.07563 (accessed on 29 June 2017).
46. Burke, K. How Many Texts Do People Send Every Day? Text Request. Blog. 2016. Available online: https://www.textrequest.com/blog/many-texts-people-send-per-day/ (accessed on 29 November 2017).
47. Sauter, M. LTE And The Number Of Simultaneously Connected Users—WirelessMoves. *Wireless Moves. Blog*. 2016. Available online: https://blog.wirelessmoves.com/2016/02/lte-and-the-number-of-simultaneously-connected-users.html (accessed on 29 November 2017).
48. Suomalainen, J. Security Monitor for 5G-Microsegments. *5G-PPP Security Enablers Documentation. 5G-ENSURE Project Deliverable 3.4.* 2017. Available online: https://5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D3.4_5G-PPP_Security_Enablers_Documentation.pdf (accessed on 28 September 2017).
49. 5G-ENSURE Project. Web Site. 2017. Available online: http://5gensure.eu/ (accessed on 28 September 2017).
50. Al-Shabibi, A.; de Leenheer, M.; Gerola, M.; Koshibe, A.; Parulkar, G.; Salvadori, E.; Snow, B. OpenVirteX. In Proceedings of the Third Workshop on Hot Topics in Software Defined Networking (HotSDN '14), Chicago, IL, USA, 22 August 2014; pp. 25–30. Available online: http://dl.acm.org/citation.cfm?id=2620728.2620741 (accessed on 23 September 2015).
51. OpenVirteX Project. Web Site. 2017. Available online: http://mininet.org/ (accessed on 28 September 2017).
52. Ryu SDN Framework Community. Web Site. 2017. Available online: https://osrg.github.io/ryu/ (accessed on 28 September 2017).
53. Mininet Team. Web Site. 2017. Available online: http://mininet.org/ (accessed on 28 September 2017).
54. Open vSwitch. Web Site. 2017. Available online: http://openvswitch.org/ (accessed on 3 October 2017).
55. Apache Kafka. Available online: https://kafka.apache.org/ (accessed on 25 August 2017).
56. Apache Spark—Lightning-Fast Cluster Computing. Available online: https://spark.apache.org/ (accessed on 25 August 2017).
57. Ailon, N.; Jaiswal, R.; Monteleoni, C. Streaming k-means approximation. In *Advances in Neural Information*; Neural Information Processing Systems Foundation: South Lake Tahoe, NV, USA, 2009; Available online: http://papers.nips.cc/paper/3812-streaming-k-means-approximation (accessed on 14 February 2017).
58. Freeman, J. Introducing Streaming k-Means in Apache Spark 1.2—The Databricks Blog. *Databrics, Engineering Blog*. Available online: https://databricks.com/blog/2015/01/28/introducing-streaming-k-means-in-spark-1-2.html (accessed on 14 February 2017).
59. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 18–21 July 1965.

60. Lloyd, S. Least Squares Quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129–137. Available online: http://ieeexplore.ieee.org/abstract/document/1056489/ (accessed on 15 February 2017). [CrossRef]
61. Core Network Dynamics. *OpenEPC*. 2017. Available online: https://www.corenetdynamics.com/products (accessed on 2 October 2017).
62. Bergmann, O. Libcoap: C-Implementation of CoAP. 2017. Available online: https://libcoap.net/ (accessed on 2 October 2017).
63. Shelby, Z.; Hartke, K.; Bormann, C. The Constrained Application Protocol (CoAP). *RFC 7252. IETF;* 2014. Available online: https://tools.ietf.org/html/rfc7252 (accessed on 2 October 2017).
64. Rosenberg, J.; Schulzrinne, H.; Camarillo, G.; Johnston, A.; Peterson, J.; Sparks, R.; Handley, M.; Schooler, E. SIP: Session Initiation Protocol. *RFC 3261.* 2002. Available online: https://tools.ietf.org/html/rfc3261 (accessed on 2 October 2017).
65. Schulzrinne, H.; Casner, S.; Frederick, R.; Jacobson, V. RTP: A Transport Protocol for Real-Time Applications. *RFC3550, IETF.* 2003. Available online: https://tools.ietf.org/html/rfc3550 (accessed on 2 October 2017).
66. The Wireshark Team. Wireshark Web Site. 2017. Available online: https://www.wireshark.org/ (accessed on 2 October 2017).
67. Borg, I.; Groenen, P.; Mair, P. *Applied Multidimensional Scaling*; Springer Science & Business Media: New York, NY, USA, 2012.
68. The Scikit-Learn Project. sklearn.manifold.MDS—Scikit-Learn 0.19.0 Documentation. 2017. Available online: http://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html (accessed on 15 September 2017).
69. Steinbach, M.; Karypis, G.; Kumar, V. A Comparison of Document Clustering Techniques. In Proceedings of the KDD Workshop on Text Mining, Boston, MA, USA, 20–23 August 2000.
70. The Scikit-Learn Project. sklearn.preprocessing.MinMaxScaler—Scikit-Learn 0.19.0 Documentation. Available online: http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html (accessed on 18 September 2017).
71. The Scikit-Learn Project. sklearn.datasets.make_blobs—Scikit-Learn 0.19.0 Documentation. Available online: http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html (accessed on 18 September 2017).
72. The Apache Spark Project. Clustering—RDD-Based API—Spark 2.2.0 Documentation. 2018. Available online: https://spark.apache.org/docs/2.2.0/mllib-clustering.html (accessed on 8 February 2018).