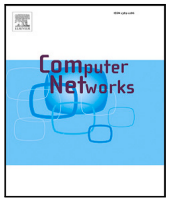
Computer Networks 249 (2024) 110518

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Design and implementation of a looking-forward Lawful Interception architecture for future mobile communication systems

Ingrid Huso [a,b,*], Marco Olivieri [a], Leonardo Galgano [a], Adnan Rashid [a], Giuseppe Piro [a,b], Gennaro Boggia [a,b]

[a] Department of Electrical and Information Engineering, Politecnico di Bari, Bari, Italy
[b] CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni, Italy

ARTICLE INFO

Keywords:
Lawful Interception
5G and Beyond 5G
End-to-end encryption
Key Escrow

ABSTRACT

Law Enforcement Agencies (LEAs) heavily rely on Lawful Interception (LI) tools to investigate criminal and terrorist activities. The growing frequency of cybercrime, terrorism-related offenses, and illegal trades in the European Union (EU) has driven LEAs to explore novel LI techniques that align with the developing 5G and Beyond 5G network architectures. Moreover, the emergence of extremely dynamic and distributed networks, the increased usage of end-to-end encryption applications, and privacy protections present limitations for traditional LI approaches. In order to provide a technological solution capable of extending the 3GPP LI standard, this paper presents a novel LI framework designed on top of the standardized 3GPP LI architecture, leveraging an inspection-friendly end-to-end cryptography mechanism (e.g., a Key Escrow algorithm) at the application layer. Moreover, the proposed Lawful Interception (LI) framework enables authorized LEAs to decrypt intercepted end-to-end encrypted data within the core network. Firstly, a security proof validates the security of the proposed LI framework under two attack scenarios. Subsequently, a proof-of-concept workstation implementation that emulates a 5G network for end-to-end data exchange and cloud-based deployment validates the suggested LI framework by affirming the LEA capabilities in decrypting intercepted data. Additionally, the system performance has been studied through experimental tests, ensuring the scalability of the conceived solution and revealing the possibility of intercepting data with mainly real-time latency without affecting the Quality of Service (QoS) experienced by the user.

1. Introduction

The European Union (EU) has witnessed a significant increment of criminal networks involved in cybercrime, terrorism-related offenses, and outlawed trades [1]. The most recent report on police-recorded offenses within the EU presents statistical insights spanning the years from 2016 to 2021 [2]. It encompasses various criminal activities across EU member states, defining occurrences such as acts against computer systems with approximately 110k cybercrime events recorded in 2021, participation in organized criminal activities, reflecting around 7.5k registered activities during the same year, and unlawful acts involving controlled drugs or preceding, accounting for over 1150k events in 2021. Moreover, it emerges an increasing level of participation by EU member states in these initiatives. For instance, the number of cybercrimes doubled across major European countries between 2018 and 2021. Therefore, Law Enforcement Agencies (LEAs) are seeking innovative and efficient LI tools that are compatible with the evolving 5G and Beyond 5G network architectures and are capable of preventing, detecting, and investigating criminal and terrorist activities.

In contrast to the conventional technologies, the 5G and Beyond 5G networks provide incomparable data rates, high channel capacity, and low latency by introducing a highly dynamic and distributed architecture with the use of emerging technologies such as Software-Defined Networking (SDN), Network Function Virtualization (NFV), network slicing, and Edge Computing [3]. To date, this technology integration is required to cope massive increase of data generated by IoT devices and applications where the majority of the data may be encrypted or in plaintext. These emerging technologies enable efficient resource allocation and on-demand network customization, making it challenging to identify precise interception points and employ advanced analytic tools for real-time interception, processing, and analysis of data within the future network infrastructure [4].

Furthermore, 5G networks use new security protocols such as enhanced encryption and random mobile identifiers [3]. Therefore, if in

* Corresponding author at: Department of Electrical and Information Engineering, Politecnico di Bari, Bari, Italy.
E-mail address: ingrid.huso@poliba.it (I. Huso).

https://doi.org/10.1016/j.comnet.2024.110518
Received 6 January 2024; Received in revised form 30 March 2024; Accepted 16 May 2024
Available online 20 May 2024

Acronym	Full text
3GPP	3rd Generation Partnership Project
5GNR	5G New Radio
5GCN	5G Core Network
ADMF	Administration Function
AMF	Access and Mobility Management Function
AUSF	Authentication Server Function
CA	Certification Authority
CC	Communication Content
CSP	Communications Service Provider
GDPR	General Data Protection Regulation
gNB	Next Generation Node B
IDBC	ID-based Cryptosystem
IM	Instant Messaging
IMSI	International Mobile Subscriber Identity
IRI	Intercept Related Information
KGC	Key Generation Centre
KPI	Key Performance Indicator
LEA	Law Enforcement Agency
LEMF	Law Enforcement Monitoring Facility
LI	Lawful Interception
LICF	Lawful Interception Control Function
LIPF	Lawful Interception Provisioning Function
MDF	Mediation and Delivery Function
NF	Network Function
POI	Point of Interception
QoS	Quality of Service
SIRF	System Information Retrieval Function
TKA	Trusted Key Authority
TF	Triggering Function
TLS	Transport Layer Security
UE	User Equipment
UPF	User Plane Function
NFV	Network Function Virtualization
SDES	Session Description Protocol Security Descriptions
SIP	Session Initiation Protocol
SDN	Software-Defined Networking
SRTP	Secure Real-time Transport Protocol
UDP	User Datagram Protocol
VoIP	Voice over IP

the past radio monitoring techniques (e.g., IMSI-catchers [5]) were used to intercept network identifiers, in the new 5G Core Network (5GCN) it is no longer feasible since the International Mobile Subscriber Identity (IMSI) is transmitted in a concealed form to protect their privacy [6]. Moreover, new-generation mobile systems are increasingly dependent on Instant Messaging (IM) and VoIP platforms (e.g., Telegram and WhatsApp), allowing, by the privacy-by-design paradigm, real-time communication and secure sharing of private information through the usage of end-to-end encryption [7]. It is a secure communication mechanism that permits only the parties involved to correctly send and receive messages since the encryption keys are only accessible to each participant and not to the service provider [8,9].

Nevertheless, while this represents a significant achievement in communication security, it makes conventional LI techniques, based on existing 3rd Generation Partnership Project (3GPP) specifications, largely ineffective [10]. Despite this, the LEAs can still intercept communication flows but the encrypted data remains fully unintelligible [11]. This introduces significant challenges for the advancement of LI methodologies, thus requiring the design and the investigation of novel technical solutions [4]. It is important to note that this challenge has gained attention from the European Commission, researchers, and security specialists [4,10], and [11].

Recent position papers such as [12,13], highlight the importance of addressing the management of LI in Beyond 5G and 6G systems and standardizing legal requirements [14]. However, it should be noted that their primary goal is not to provide any original or effective methodology for solving this problem. Meanwhile, a machine learning-based LI architecture, as described in [15], has been designed to analyze and classify audio and video content. Nonetheless, it does not offer the possibility to decrypt the multimedia flows, as well as to deliver them to LEAs. As a result, the usage of end-to-end encryption in an ever larger amount of applications highlights the importance of introducing more sophisticated techniques supporting effective LI features.

To bridge this fundamental gap, our work[1] presented herein provides the following main scientific contributions:

- We present a novel LI framework offering new interception capabilities on top of the existing 3GPP standardized architecture. The proposed LI framework leverages a secure configuration and usage of an inspection-friendly end-to-end cryptography scheme (i.e., Key Escrow algorithm) at the application layer and allows authorized LEAs to decipher end-to-end encrypted data intercepted (via conventional LI procedures) in the core network. Here, data privacy is guaranteed against the mobile operator, which is still unable to guess intercepted contents because encrypted. Moreover, the security proof study demonstrates the ability of the proposed LI framework to resist two adversarial scenarios.
- We present a proof-of-concept implementation of the proposed LI framework, based on the Linux-based Docker containers, emulating a 5G network via Open5Gs and UERANSIM environments. Herein, we use the OpenLI software to ensure the standard-compliant LI implementation by employing four containers representing the entities of the LI framework. Our implementation, using Python scripts and its cryptographic libraries, demonstrates functionalities such as end-to-end encrypted data exchange, data interception, and decryption through Key Escrow mechanisms at the application layer.
- We evaluate the performance of the proposed approach by considering two different use cases: an end-to-end data exchange (i.e., encrypted end-to-end file exchange) and a cloud-based deployment (i.e., VoIP service). The obtained results validate the effectiveness and reveal the real-time-like latency performances and scalability of the proposed LI framework.

The rest of the paper is organized as follows. Section 2 specifies some background concepts on LI and Key Escrow and highlights the main challenges that drove our work. Section 3 presents the proposed methodology, offering comprehensive information on the integrated cryptographic algorithm and its security proof, as well as the designed communication protocol and associated procedures. The proof-of-concept implementation is presented in Section 4. We explore the considerable potential of the proposed LI framework through experimental testing in Section 5. Finally, we conclude the paper and draw future research activities in Section 6.

2. Background and motivation

This Section explains the technicalities of the standardized 3GPP LI architecture, presents background concepts on End-to-End encryption techniques, and describes the state of the art on Key Escrow schemes.

[1] This work represents a substantial extension of a preliminary contribution previously presented by the same author in a recent conference paper [16].

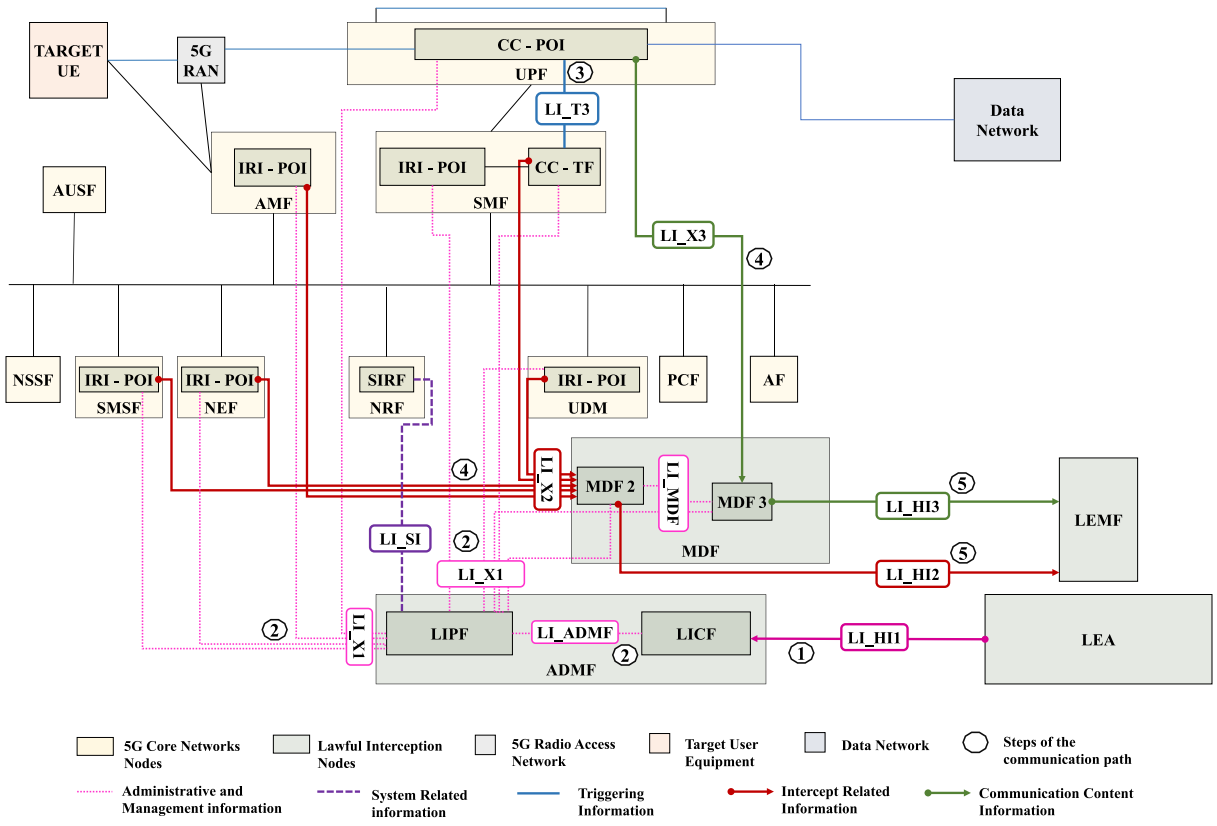


Fig. 1. 5G 3GPP Lawful Interception architecture.

2.1. Lawful interception

The LI refers to the technological methods employed by Communications Service Providers (CSPs) to collect, retain, and transmit communication data to law enforcement databases [17]. The 3GPP Technical Specifications on LI provides: (i) LI requirements in TS 33.126 [17], (ii) LI architecture and functions in TS 33.127 [18], and (iii) LI protocol and procedures in TS 33.128 [19].

Fig. 1 illustrates a high-level description of the 5G 3GPP LI architecture, highlighting LI nodes and interfaces. Within this illustration, the communication paths proceed through five main steps, which are as follows:

- ***Step 1.*** Given a targeted User Equipment (UE) which needs to be intercepted, the LEA submits a valid warrant to the CSP through the LI_HI1 interface, which in turn starts all the required standard procedures [17].
- ***Step 2.*** Herein, in accordance with [18], the Administration Function (ADMF), by exploiting the LI_ADMF and all LI_X1 interfaces are responsible for the administrative and management functions of the LI capability within the CSP. These functions encompass the provisioning, modification, and deactivation of Point of Interception (POI), Triggering Function (TF), and Mediation and Delivery Functions (MDFs).
  Specifically, the ADMF comprises two main logical sub-functions, communicating via the LI_ADMF interface.
  First, the Lawful Interception Control Function (LICF) manages the entire life cycle of a warrant while acting as the central repository for all sensitive information and LI configuration data. Additionally, it holds the ultimate responsibility for all decisions made within the LI system.
  Second, the Lawful Interception Provisioning Function (LIPF) serves as a secure intermediary that enables the LICF to interact with the LI modules that are necessary for the CSP network to function. Indeed, it is in charge of interacting with the System Information Retrieval Function (SIRF), which gives interface system-related information via the LI_SI, so that the latter may carry out the steps required to set up and sustain interception of the target service.
  Indeed, the LIPF performs a passive function during this step. By routing LI_X1 communications from and to the LICF, or an active function by receiving triggering information and passing the trigger to the relevant POI.
- ***Step 3.*** This step begins when the relevant POI, located in the User Plane Function (UPF), is triggered via the LI_T3 interface enabling it to (i) detect the target communication, (ii) extract Intercept Related Information (IRI) or Communication Content (CC) from the target, and (iii) deliver the output to the MDF [19].
- ***Step 4.*** In this step, the architecture provides multiple POIs distinguished into two groups based on the type of information they transmit to the MDF, which comprises two modules (i.e., MDF2 and MDF3). Therefore, the IRI-POI delivers IRI information through the LI_X2 interface to the MDF2, while the CC-POI delivers CC data over the LI_X3 to the MDF3.
- ***Step 5.*** At this point, the MDF generates the IRI and CC messages from the MDF2 and MDF3 and delivers them, via the LI_HI2 and LI_HI3 interfaces, respectively, to the Law Enforcement Monitoring Facility (LEMF) [19]. Finally, the LEA easily accesses the intercepted traffic.

2.2. End-to-end encryption

As previously anticipated in Section 2, one of the most relevant challenges regards *encryption and privacy*. The widespread implementation of encryption mechanisms in 5G and Beyond 5G networks poses substantial obstacles to the LI process. People increasingly depend on applications such as Skype, Zoom, Telegram, WhatsApp, or similar

applications, which generate extensive multimedia content, including text, voice, and video. Consequently, the demand for robust sensitive data protection systems has risen to protect this vast multimedia content. One way to achieve this is to put in place an end-to-end encryption system that does not rely on any online services or centralized infrastructure. Indeed, more VoIP and IM applications claim to support end-to-end encryption which guarantees that only the sender and the intended receiver can decipher the contents of a message [7].

In the realm of secure online communication systems and private chat applications, the off-the-record (OTR) protocol emerged to facilitate end-to-end encryption [20]. Despite being integrated as a plugin for widely used IM clients such as Pidgin, its limited adoption can be attributed to usability issues [21]. Increased consciousness of privacy concerns emerged after the Snowden revelations. As a result, new encrypted messaging systems have evolved to address end-to-end encryption problems by expanding and adopting the OTR protocol [7]. To provide both end-to-end encryption and advanced security features, such as forward secrecy and future secrecy, Open Whisper Systems introduced Signal, a groundbreaking end-to-end encryption protocol that supports both synchronous and asynchronous communication settings. The Signal protocol requires a key-distribution server to maintain user identities and ephemeral keys, as it functions in synchronous and asynchronous messaging situations [22].

Currently, the majority of end-to-end encryption applications either employ the Signal protocol (e.g., Signal and WhatsApp) or use Signal-like proprietary protocols (e.g., Telegram and Zoom) [7]. For example, the WhatsApp end-to-end encryption requires a user to initiate a voice or video connection by creating encrypted sessions with each of the receiver devices, and after the call is initiated, Secure Real-time Transport Protocol (SRTP) is used to protect it by using master secret keys created for each receiver device [23]. Whereas, the Telegram end-to-end encryption functionality is implemented in one-to-one chats and calls using its proprietary protocol, known as the MTProto protocol. In this protocol the cryptographic keys are exchanged via the Diffie–Hellman protocol and the participating devices exchange these keys after establishing a Secret Chat [24].

Even if the adoption of end-to-end encryption significantly enhances the security and privacy of communications, it simultaneously renders the interception of the communication more challenging [9]. In case of end-to-end encryption, in fact, the intercepted traffic can be interpreted by LEAs just as a string of bits with limited information. In this context, effectively managing the trade-off between privacy or security and the requirements of authorized interception becomes crucial. Therefore, the mitigation of these challenges demands the development of resilient decryption capabilities and the establishment of collaborative frameworks between telecommunication service providers and LEA.

2.3. Key Escrow

Generally speaking, Key Escrow represents a technique that helps in recovering the secret key used for application encryption and, when specific criteria are met, assists the authorized entities (e.g., the LEA in our case) in decrypting the ciphertext [25].

The Clipper Chip was proposed by the United States government in the 1990s as an initial effort to build a key escrow system [26]. Herein, the Skipjack symmetric encryption was employed and the encryption keys were partitioned into distinct components and securely entrusted to various government entities [27]. Nonetheless, the Clipper Chip faced intense criticism and censure owing to concerns about its susceptibility to security flaws and the inherent hazards of unauthorized access to the escrowed keys, weakening its usefulness and public trust [28]. To overcome the previous issues and strike a balance between enabling lawful interception and mitigating the potential for unauthorized access, an alternative strategy for Key Escrow entails the engagement of a trusted third-party entity responsible for preserving the decryption keys on behalf of users [29]. To facilitate this form of Key Escrow, various protocols have been suggested, including the ones in [25,30], and [31], which aim to provide the necessary framework for effective implementation and management.

Apart from these contributions, the adoption of Key Escrow techniques in the context of LI has not received the deserved attention in recent years. The main reason refers to the native design principle of the related interception approach: the introduction of the General Data Protection Regulation (GDPR) led to the prohibition of previous Key Escrow schemes that operated on SIM private keys, as users were unaware of the voluntary backdoors [32]. Remarkably, to the best of the authors' knowledge, there has been no attempt to employ a Key Escrow system at the application level for LI purposes.

In contrast, we believe that Key Escrow schemes, applied to end-to-end cryptography, may achieve a compromise between the need for individual privacy and the lawful requirements of government agencies to conduct surveillance or interception activities for criminal investigations [28].

Indeed, it underscores the need for further research and development in addressing the evolving landscape of privacy regulations and technological advancements and offering valuable insights into the potential integration of Key Escrow mechanisms within application frameworks for enhanced LI capabilities.

3. The proposed methodology

This Section aims to propose a feasible technical solution that fosters further discussions on defining inspection-friendly end-to-end encryption schemes to address the LI challenges. It introduces a novel LI framework that enhances interception capabilities by adding new features on top of the conventional 3GPP standardized architecture. This enhancement is achieved through the secure configuration and utilization of a Key Escrow cryptographic scheme at the application layer, thereby enabling LEAs to decrypt end-to-end encrypted data intercepted in the core network.

3.1. Design principles

The proposed LI framework, illustrated in Fig. 2 has been designed starting from the following two main hypotheses.

(1) Standard-compliant hypothesis. In accordance with the guidelines provided by 3GPP [18], the mobile network infrastructure consists of the Next Generation Node B (gNB) that facilitates wireless connectivity for UEs through the 5G New Radio (5GNR) interface, as well as the 5GCN. The mobile network operator possesses control over the network infrastructure, enabling the end-user's identification through the International Mobile Subscriber Identity (IMSI) and determining the corresponding UPF within the 5GCN. Notably, the UPF also serves as the hosting entity for the POI, which owns the capability to intercept specific communications as elaborated upon below.

(2) High level definition of the proposed LI framework. The traffic generated or received by the UE results in encrypted application data, meaning that the intercepted CC potentially comprises a series of encrypted data. Thus, the technical approach assumes that the end-to-end application traffic is secured through a Key Escrow system, which assists authorized entities such as the LEAs in decrypting the ciphertext [29].

Indeed, the conceived LI framework is defined on top of the conventional 3GPP scheme reported in Fig. 1. Specifically, it does not require new 3GPP entities and it considers the following four main entities:

- **Subscribers.** Two users (i.e., UE A and UE B) enabling secure communication via end-to-end encryption.
- **Law Enforcement Agency (LEA).** An authorized enforcement entity that, under the legislation, requests the content of the communication and gets IRI and CC from the CSP.

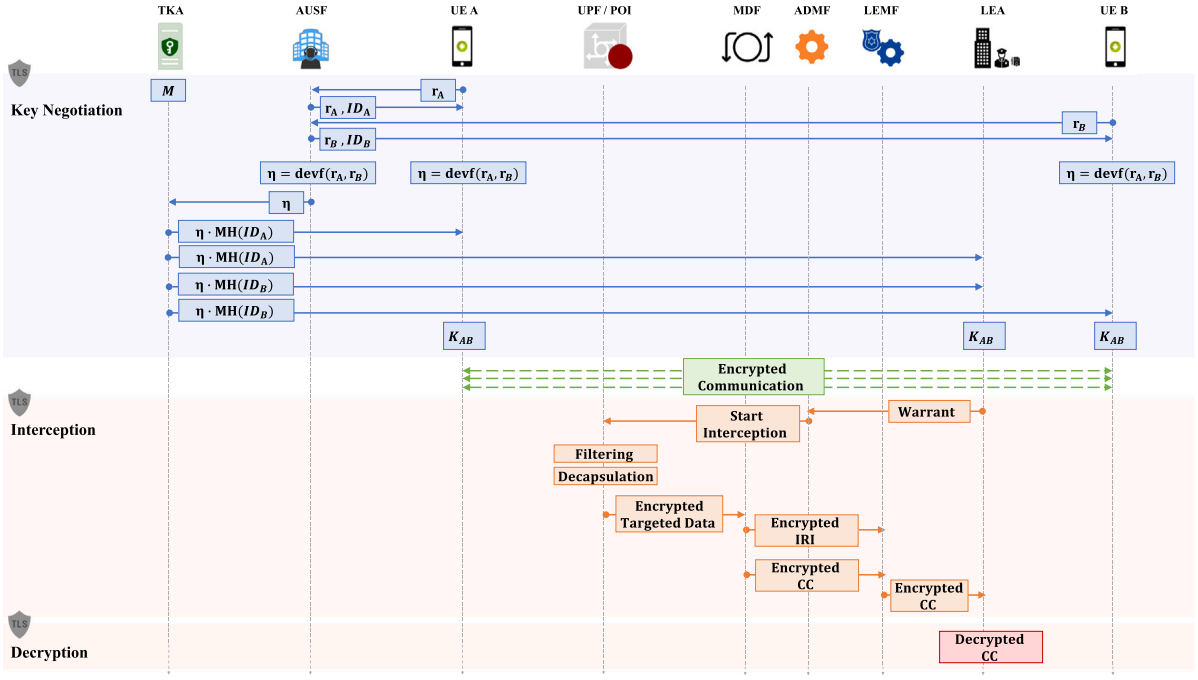


Fig. 2. Technical workflow.

- **Authentication Server Function (AUSF).** A responsible component of the 5GCN for subscribers' identity verification. In response to the LEA request, it sends to the LEA the appropriate decryption material and interception-related information and provides the application encryption material to the subscribers.
- **Trusted Key Authority (TKA).** A fully trusted third party that requests and provides encryption keys for communication sessions to the AUSF, LEA, and subscribers.

Without loss of generality, the proposed solution leverages the Key Escrow algorithm presented in [29], constructed on an ID-based Cryptosystem (IDBC), and investigates, for the first time, its adoption on LI tasks within Beyond 5G systems. Specifically, the proposed LI framework builds upon the conventional 3GPP LI architecture [18] by introducing a new entity, namely TKA. This entity, acting as a fully trusted third party with the same degree of trustworthiness as a Certification Authority (CA), assumes system configuration responsibility by addressing both application security setups and application session key material generation. It is necessary to emphasize that the TKA only retains its secret master key and does not store the keypairs of any registered user.

Moreover, it is important to highlight that the conceived LI framework requires an end-to-end encrypted application provider to agree on the specifically defined algorithm as a key-exchange solution.

To ensure clarity, the conceived LI framework outlines above the standardized 3GPP LI architecture [18], which independently manages the aspects related to mobility behavior. In particular, in line with the standard [18], each piece of mobility information (e.g., location and cell IDs) is stored within the IRI content. Firstly, when a target UE connects to the 5G network in the registration procedure, the IRI-POI in the Access and Mobility Management Function (AMF) creates the registration xIRI, which contains information on the registration mobility update. Subsequently, the location update xIRI is produced each time the IRI-POI in the AMF determines that the targeted UE's location has changed due to UE mobility or when the AMF sees target UE location data while performing a service operation. Furthermore, if the information in the AMF includes one or more cell IDs, all of them must be transmitted to the LEMF whenever location reporting is activated at the AMF.

3.2. Technical details

This section better describes the conceived LI framework as depicted in Fig. 2. To ensure clear understanding, it is important to note that the interaction between the TKA and 5GCN or 5GNR nodes is protected via the Transport Layer Security (TLS) protocol.

Moreover, based on the two premises introduced in Section 3.1, system configuration and interception operations can be described through three main phases: *key negotiation*, *interception*, and *decryption*.

1. ***Key Negotiation Phase*:** This phase relies on the involvement of both the mobile operator and TKA. In this context, the formulated LI framework performs most of the application-level cryptographic operations by introducing hash functions, bilinear pairings, and derivative functions denoted as $\mathcal{H}(\cdot)$, e(·), and η respectively. The TKA owns a master secret key and computes the UEs public/private key pairs based on their unique identity. Moreover, based on the Key Escrow algorithm presented in [29], since pre-shared keys are distributed between the AUSF and two UEs, the AUSF securely transmits and receives nonces to the UEs. Thus, the two UEs are equipped to independently compute the derivation functions, obtain the application session key, and protect the communication using end-to-end encryption at the application layer (i.e., $k_{AB} = e(\eta \cdot M\mathcal{H}(ID_A), \mathcal{H}(ID_B))$ and $k_{BA} = e(\mathcal{H}(ID_A), \eta \cdot M\mathcal{H}(ID_B))$). At the same time, the AUSF calculates the derivation function and shares it with the TKA, which, in turn, forwards it to the authorized LEA. Thus, the LEA owns the cryptographic material for deriving the same application session key while ensuring that users remain unaware of lawful interception activity. Fig. 2 presents the detailed cryptographic operations. Please refer to the Appendix A and to [29] for in-depth details about the summarized cryptographic scheme.
2. ***Interception Phase*:** During this phase, the LEA issues a valid interception warrant to the ADMF. The ADMF validates the warrant and subsequently grants permission to commence the interception procedure. Herein, the POI decapsulates and filters the targeted GTP data. Specifically, after the ADMF validates the warrant, the POI investigates the traffic passing through the UPF. In line with the *Standard-compliant hypothesis*, the packets

containing the data exchanged between the two UEs in the end-to-end encrypted communication are encapsulated according to the following protocol structure: IP over GTP over TCP over IP. Indeed, the POI performs a decapsulation operation on each packet to obtain the raw end-to-encrypted application data. Subsequently, the LEMF entity collaborates with the MDF to collect precise information about the targeted communication from POI, including IRI and encrypted CC.

3. **Decryption Phase:** During this phase, the LEA gains the ability to decrypt the previously received encrypted CC. This decryption process involves the utilization of the application session key, denoted as $k_{AB} = e(\eta \cdot M\mathcal{H}(ID_A), \mathcal{H}(ID_B))$.

To ensure clarity, the conceived LI framework is established on the existing 3GPP architecture and does not modify the established 5G security protocol. The above-described cryptographic procedures are meant to be done at the application layer and do not involve the UE SIM, which has limited computing capabilities. Moreover, it is important to remark that since the application session key is derived from random numbers exchanged in every new negotiated communication, the LEA cannot reuse the same key material for intercepting other communication sessions.

3.3. Security proof and threat analysis

This Section provides the security proof for the designed LI framework, considering the security requirements related to the communication protocols and cryptographic techniques. It is important to note that [29] examines the security proof of the selected Key Escrow technique in terms of cryptographic operations. Concerning the protocol security analysis for the proposed framework, the developed LI framework incorporates a widely recognized security building block known as TLS. Its security has been previously established and fully defined in the reference contributions given below, and it remains independently created. As a result, the following proves the security of each employed security requirement:

- **Secure End-to-End Communication:** Through the use of the TLS (i.e., TLS version 1.3) protocol, the proposed LI framework ensures the establishment of a secure end-to-end channel communication between each non-5G entity and 5G standardized network architecture. In particular, it facilitates mutual authentication and data secrecy. Furthermore, it allows the communication network to be resistant against Man-in-the-Middle (MITM) attacks. Since TLS is a well-known and widely used security protocol, [33,34], and [35] have already investigated its security proof.
- **Subscriber Non-Engagement:** This security requirement ensures that subscribers cannot determine whether their communication is being monitored since they do not take part in key escrowing which mainly involves the TKA and the AUSF. The work in [29] provides formal proof of it.
- **Warrant Validity:** This security requirement relates to the failure of an interception in an unauthorized session and to the prevention of a replay attack. Specifically, as detailed in Appendix A, the LEA receives from the TKA the cryptographic material (i.e., τ_2) for calculating the session key after verifying the previously submitted specific warrant. Moreover, since the Key Escrow algorithm selects nonces r_A and r_B randomly in each session and the application session key is derived from a function involving these nonces (i.e., $\eta = devf(r_A, r_B)$), each application session will introduce a different session key not managed from the submitted warrant, as demonstrated by [29].
- **Key Escrow Effectiveness:** The generic PKI-based Key Escrow models need the TKA to store a large number of public key pairs, while the proposed LI framework only requires the storage of the master key, which is always kept secure and never delivered, in line with [29].

Moreover, to ensure the compliance with LI specifications and standards (i.e., [17,18], and [19]) which allow only authorized LEA with a valid warrant to intercept the communication, this Section aims at studying the security of the proposed LI framework under two attack scenarios involving the presence of a malicious user (i.e., unauthorized LEA) trying to eavesdrop the secure communication, in line with [29].

Adversarial scenario 1: absence of a valid warrant. Let UE A and UE B be the subscribers willing to initiate an end-to-end encrypted communication and let the eavesdropper E be the malicious user trying to intercept the encrypted communication. The procedure pursues the following steps in line with the algorithm described in Appendix A:

1. The UE A sends the chosen random number r_A and its signature $sign_A(r_A)$ to the AUSF.
2. The AUSF verifies r_A and the signature $sign_A(r_A)$ of UE A and forwards them to the UE B.
3. UE B, in turn, verifies the UE A signature, generates its random number r_B, and delivers it together with its signature $sign_B(r_B)$ to the AUSF.
4. The AUSF proves r_B and the signature $sign_B(r_B)$ of UE B and forwards them to the UE A.
5. The two subscribers now compute their application session key $k_{AB} = e(devf(r_A, r_B) \cdot P_A, p_B)$ and $k_{BA} = e(p_A, devf(r_A, r_B) \cdot P_B)$, respectively, and start their end-to-end encrypted communication (please refer to Appendix A for detailed description.).
6. The eavesdropper E tries to intercept the communication, but it does not have any related cryptographic material from which to retrieve the application session key because it did not present any warrant to let the TKA generate and forward it. Specifically, it does not have any information about the derivation function and fails to compute any correct application session key for decrypting the communication between UE A and UE B.

Adversarial scenario 2: expired session. Let UE A and UE B be the subscribers that already had an end-to-end encrypted session correctly intercepted by an authorized LEA, and let the eavesdropper E be the malicious user capturing the application session key used for the above communication session (i.e., $k_{AB} = e(devf(r_A, r_B) \cdot P_A, p_B)$). Assuming now that UE A and UE B start a new communication session, they compute the new application session key as follows:

1. The UE A forwards the chosen random number r'_A and its signature $sign_A(r'_A)$ to the AUSF.
2. The AUSF verifies r'_A and the signature $sign_A(r'_A)$ of UE A and sends them to the UE B.
3. UE B, in turn, proves the UE A signature and generates its random number r'_B and delivers it together with its signature $sign_B(r'_B)$ to the AUSF.
4. The AUSF verifies r'_B and the signature $sign_B(r'_B)$ of UE B and onwards them to the UE A.
5. The two subscribers calculate their application session key $k'_{AB} = e(devf(r'_A, r'_B) \cdot P_A, p_B)$ and $k'_{BA} = e(p_A, devf(r'_A, r'_B) \cdot P_B)$, respectively, and start their end-to-end encrypted communication (please refer to Appendix A for detailed description.).
6. The eavesdropper E attempts to capture the communication, but it has the wrong application session key derived from r_A and r_B, which is different from the newly defined one and fails to decrypt the communication between UE A and UE B. Specifically, even having a previous session key and considering $P_A = M\mathcal{H}(ID_A)$, retrieving the value of $devf(r'_A, r'_B)$ from $devf(r'_A, r'_B) \cdot M\mathcal{H}(ID_A)$ is not possible due to the computational infeasibility of the Elliptic Curve Discrete Logarithm Problem (ECDLP), as proved in [36].

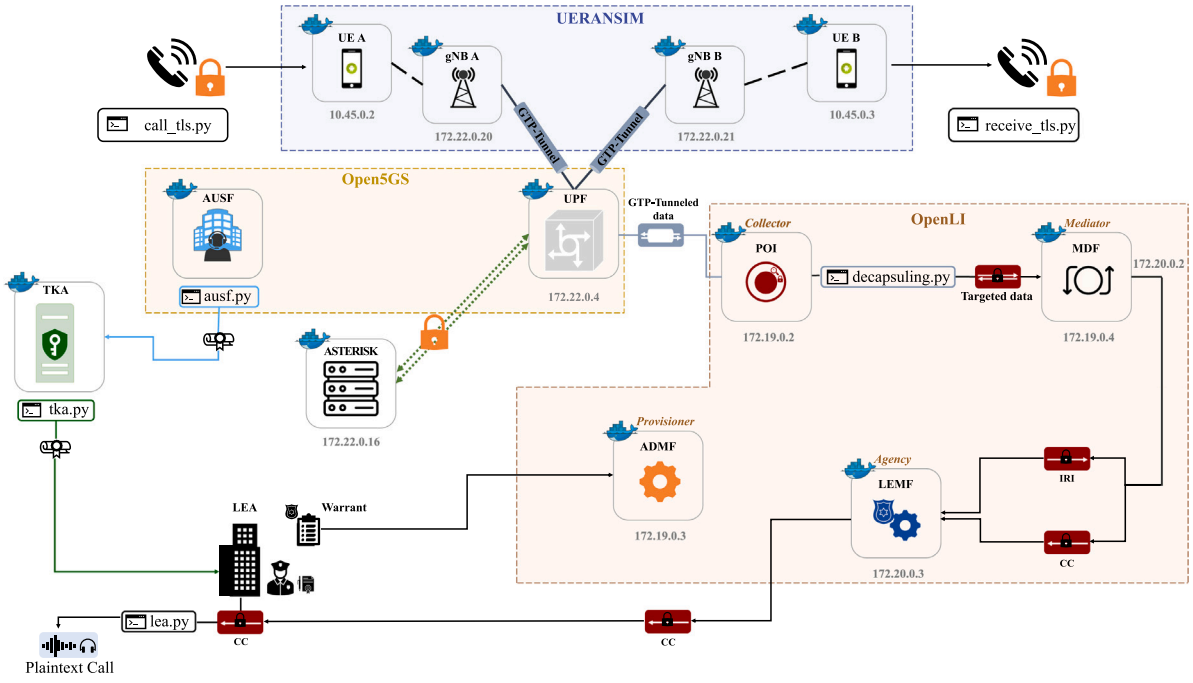


Fig. 3. VoIP services implementation setup.

4. Proof-of-concept implementation

This Section presents a proof-of-concept of the proposed LI framework which is implemented for end-to-end data exchanges (i.e., encrypted end-to-end file exchange) and cloud-based deployments (i.e., VoIP services) to prove the effectiveness of the proposed solution. Precisely, the implemented LI framework offers the opportunity to achieve the following functionalities:

- Enabling two UE devices to exchange end-to-end encrypted data across the 5GCN
- Allowing the LEAs to intercept and access downlink end-to-end encrypted data.
- Facilitating decryption of the application intercepted data through Key Escrow mechanisms.

The testbed is deployed on a workstation with an Intel(R) Core(TM) i5-9400 CPU @ 2.90 GHz processor and 16 GB of RAM. It hosts Linux-based Docker containers (i.e., Ubuntu 20.04), with a dedicated container for each 5G entity network. For emulating the 5GNR and the communication between UEs and the gNB, *UERANSIM* is installed. Simultaneously, *Open5gs* is configured to emulate the 5GCN. The *OpenLI* framework is deployed into a Docker-based environment to ensure a standard-compliant LI implementation. More specifically, four containers such as Provisioner, Collector, Mediator, and Agency are used to emulate the ADMF, POI, MDF, and LEMF.

To effectively meet the requirements of each involved node, we design and configure the network architectures illustrated in Figs. 3 and 4. This process includes executing individual environments, assigning dedicated network interfaces, and establishing their interactions. To enhance clarity, the Python scripts, by using the libraries listed in the Table 1, implement the main functionalities of each participating entity as well as the LI framework cryptographic operations.

Upon the establishment of the 5G network, the testing process begins with the exchange of cryptographic material between the AUSF and TKA using the `ausf.py` script. Subsequently, the `tka.py` script forwards this cryptographic material to the LEA for the session key computation. Once the key-negotiation phase is completed, the first deployment of the proposed framework involves the implementation

Table 1
List of software and tools.

5G network and Lawful Interception	Software
Access Network	UERANSIM
5G Core Network	Open5gs
Lawful Interception	OpenLI
End-to-end communication	Netcat
VoIP services	Asterisk server and PJSIP library
Cryptographic Operation	**Adopted libraries**
Hash function	Hashlib, libnum
Key derivation function	PyCryptodome
Encryption, decryption	PyCryptodome
Bilinear paring	Tate_bilinear_pairing
Post-processing step	**Software**
Interception	OpenLI and libtrace
Packet decapsulation	Scapy
Reassembly	TCPReassembly

of *VoIP services implementation* using the Asterisk server, as depicted in Fig. 3. Without loss of generality, the implemented proof-of-concept leverages on a key-exchange solution agreement between the VoIP provider and the designed LI framework. Specifically, the two users equipment (i.e., UE A and UE B) are registered through the `pjsip` library by starting a TLS session into the server to make or receive VoIP calls. In this way, the VoIP call will be encrypted using *SRTP/Session Description Protocol Security Descriptions (SDES)* as a key-exchange solution. After the TLS handshake, encrypted Session Initiation Protocol (SIP) messages traverse the network while the SRTP stream is encrypted by the algorithm selected during the SRTP/SDES key exchange system. Consequently, the UE A, through the `call_tls.py` script, utilizes the obtained session key to encrypt and authenticate the initial SRTP stream. Meanwhile, the UE B, employing the `receive_tls.py` script, can respond or terminate the incoming call.

Alternatively, a second scenario, illustrated in Fig. 4, involves the implementation of encrypted end-to-end file exchange. Herein, the first UE encrypts a file containing plaintext media content using the `encr_ueA.py` script and obtains the ciphertext file. The latter is then forwarded to the second UE using the `exchange_data.py` script, employing the netcat package.

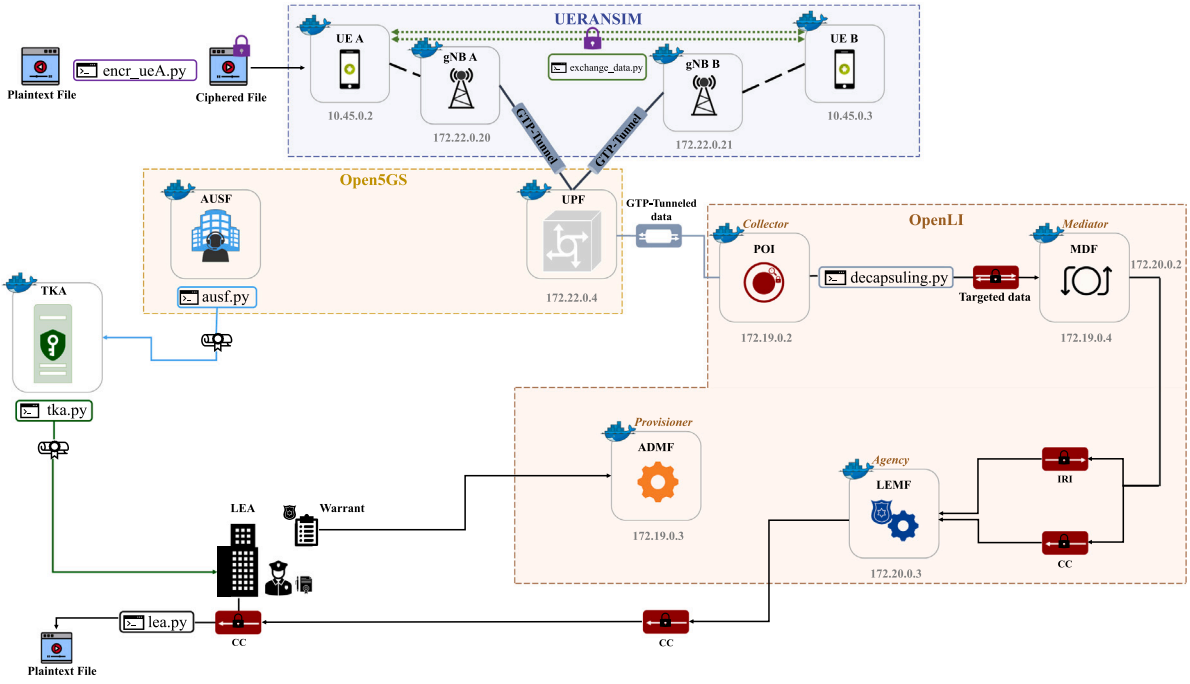


Fig. 4. End-to-end file exchange implementation setup.

Meanwhile, the interception phase starts when the LEA sends a warrant containing all the interception requirements for the downlink interception. In detail, HTTP requests with JSON files are sent to the Provisioner via the REST API. The main JSON file defines some of the warrant characteristics (e.g., *LEA ID, LEA IP and ports, interception ID, targeted UE IP, targeted UE mobile operator, and session ID*). Thus, the Provisioner can accept the interception request and activate the Collector to start the interception by letting it access the above-described JSON file. During the downlink phase, GTP-encapsulated SRTP data and encrypted data traverse the 5GCN in the first and second deployments, respectively. Herein, using the `decapsulating.py` script, which utilizes Scapy library, the collector filters and decapsulates the GTP traffic and obtains the encrypted TCP payload. Categorically, it performs a decapsulation operation on each packet to read the destination IP address of the GTP payload. If there is a match between the IP address of the analyzed GTP payload and the target IP address specified in the warrant. Moreover, the captured GTP payload is transmitted to the Collector by adding an appropriate Ethernet 802.3 header. Thus, the Collector captures the whole traffic, and by using *OpenLI* services it identifies the encrypted target data and forwards all corresponding packets to the Mediator. The Mediator receives and uses the packet-level tracing environments such as `tracepktdump` and `tracesplit` and splits encrypted targeted data in IRI and CC payload (see detailed packet inspection in Fig. 5). Later it forwards them to the Agency within the standardized interfaces (i.e., HI2 and HI3). Finally, during the decryption phase, by running the `lea.py` script and its *decryption* function, the LEA can decode the SRTP flow or decipher the target traffic and acquire the clear VoIP conversation or obtain the plaintext media file, respectively.

5. Performance evaluation

This Section investigates the significant potential of the proposed LI framework through experimental tests. Specifically, it analyzes the impact of (i) *several processed packets*, (ii)*the durations of VoIP call and the sizes of media files by measuring the latency involved in the LI procedure*, and (iii) *the deployment of the proposed LI framework on the experienced user Quality of Service (QoS)*. For this reason, four *Key Performance Indicators (KPIs)* are considered for the real-time LI latency and one *KPI* for the experienced user QoS, as follows:

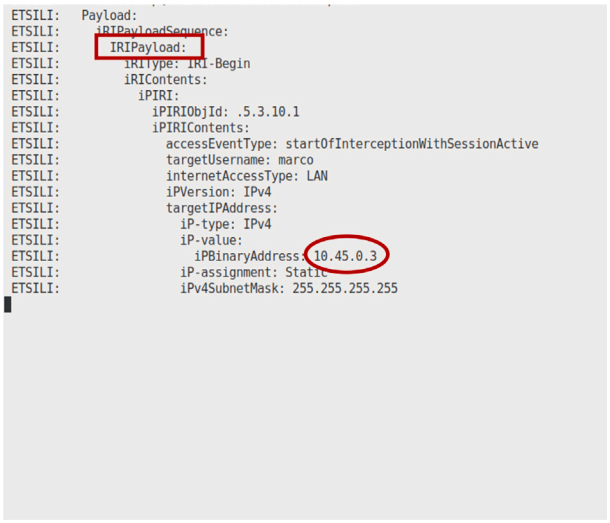
1. **UPF Acquisition Latency:** it defines the starting point of the interception procedure, and it specifies the time duration for each packet to arrive at the UPF.
2. **POI Capturing Latency:** it specifies the time duration required for each packet to be captured by the Collector.
3. **LEMF Collecting Latency:** it is the time duration in which each targeted packet is delivered to the Agency.
4. **End-to-end LI Latency:** it is the time required to process each packet during the interception process. We consider it as the sum of the above three metrics.
5. **End-to-end User Latency:** it defines the end-to-end delay experienced by each packet delivered by UE A in reaching UE B.

5.1. LI for real-time VoIP call

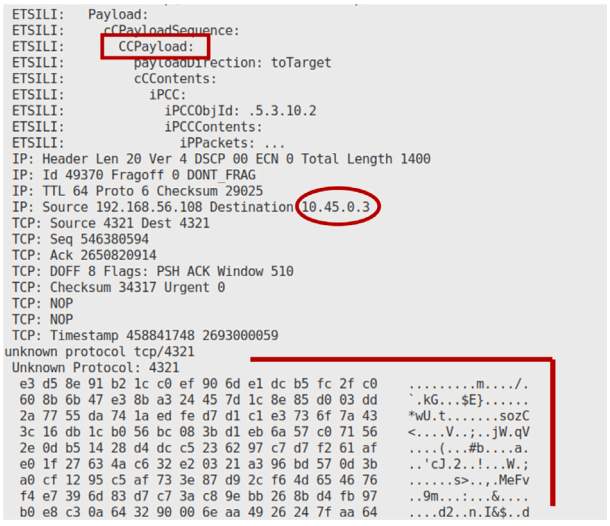
Tests examine four VoIP call conversations of varying time durations (15, 30, 45, and 60 s). The used VoIP simulation setup is described in Section 4. Each run is repeated 10^2 times over multiple seeds, with an average of the KPI measurements.

The initial evaluation assesses the impact on the number of processed packets. Fig. 6, displays for each SRTP packet, the average latency into the four phases within 30 Sec VoIP call (i.e., 1550 SRTP packets). Herein, it is important to highlight that the mean latency experienced by each packet between the *UPF Acquisition Latency* and *LEMF Collecting Latency* is of a microsecond order, emphasizing the potential of processing real-time interception activities. Additionally, the *UPF Acquisition Latency*, *POI Capturing Latency*, and *LEMF Collecting Latency* consistently hover around 20 ms, where only the 1% of packets reach higher latency times (i.e., between 30 ms and 40 ms). Moreover, analyzing the *End-to-end LI Latency*, each targeted SRTP packet reaches the LEA in less than 0.07 s, demonstrating the capacity to manage real-time interceptions even during real-time VoIP calls.

Secondly, we analyze the influence of VoIP call duration on the entire interception procedure. Fig. 7 shows the average and the statistics information of the End-to-end LI Latency per packet for the four different VoIP call durations. Specifically, it illustrates the 25th, 50th, and 75th percentiles, as well as the lowest and highest values of the



(a) IRI.



(b) CC.

Fig. 5. Intercepted Data.

End-to-end LI Latency, reached by each packet during the LI framework tests and it envisages how there is not any notable difference between the four VoIP call durations. In reality, the End-to-end LI Latency for each packet is typically between 55 ms and 65 ms. Furthermore, it also shows that the average End-to-end LI Latency per packet stays within 60 ms during the four VoIP calls, proving the scalability of the proposed methodology.

5.2. LI for end-to-end file exchange

By exploiting the end-to-end file exchange implementation setup presented in Section 4, tests consider four media files of different sizes (i.e., 10 KB, 10^2 KB, 10^3 KB, and 10^4 KB), where each run is repeated 10^2 times over multiple seeds and average KPI measurements are collected.

The initial test evaluates the impact on the number of processed packets. Fig. 8 reveals the average latency for each packet across the four phases within a specific number of packets (i.e., 7000 packets). The comprehensive results show that the differences between the main three phases are negligible, as no single phase significantly affects the End-to-end LI Latency more than the others.

In detail, the End-to-end LI Latency for each packet consistently hovers around 0.25 ms, with a small percentage of packets reaching latency times of 0.5 ms. However, it is evident that each targeted packet arrives at the LEA mostly in less than 0.5 ms by highlighting the opportunity and ability of the proposed solution to process real-time interceptions during end-to-end file exchanges.

Secondly, we evaluate the influence of the exchanged file sizes on the entire duration of the end-to-end file exchange interception. Fig. 9 shows the statistical data of the End-to-end LI Latency phase per packet as a function of the four file sizes. Herein, the average End-to-end LI latency per packet and the lowest and highest values are displayed, together with the 25th, 50th, and 75th percentiles. It should be noted that the minimum and maximum latencies reached by each packet are strongly dependent on the file size. Indeed, in small files (i.e., 10 KB) the latency varies between 0.12 ms and 0.14 ms, while in the heavier ones, it ranges between 0.08 ms and 0.5 ms. Additionally, the packet average End-to-end LI Latency verifies a dependency on the file size since it grows as the size of the exchanged file increases. In detail, when passing from a file size of 10 KB to 10^4 KB, the average latency exhibits an increase of two orders of magnitude. Nevertheless, Fig. 9 displays that even with a media file of 10^4 KB, the average packet End-to-end LI Latency at the LEA side is just over 0.25 ms by ensuring that each packet is averagely processed in real-time by the LEA.

5.3. LI impact on the user QoS

This section aims to evaluate how the deployment of the proposed LI framework affects the experienced user QoS by studying the behavior of the proposed LI framework proof of concept by activating and deactivating the LI services in both real-time VoIP calls and file exchange scenarios.

Specifically, the tests examine four VoIP call conversations of different time durations (15, 30, 45, and 60 s) and four media files of different sizes (i.e., 10 KB, 10^2 KB, 10^3 KB, and 10^4 KB). The VoIP and file exchange simulation setups used are described in Section 4. Each run is repeated 10^2 times over multiple seeds, with an average of the KPI measurements.

In particular, Fig. 10 illustrates the end-to-end user latency experienced by each packet delivered from UE A to UE B during a 30-second VoIP call. It is noticeable that there is not a significant variation in the end-to-end user latency in terms of delay generated within the proposed LI framework. Indeed, for almost all packets, the time each packet takes to reach the UE B device when LI services are not going on is equivalent to the time it takes when the proposed LI framework is active. In detail, by adding LI services, each packet encounters an average delay of around 38 microseconds. The reality is that although the proof-of-concept employs exclusive containers, the tests run on a single workstation, which may have influenced and caused the above-mentioned minor delay.

Meanwhile, Fig. 11 depicts the end-to-end latency experienced by each packet delivered from UE A to UE B for transmitting a 10^3 KB exchanged file. Here, the suggested LI framework implementation results in a slight increase in the packets' end-to-end latency. In particular, using LI services, each packet experiences an average delay difference of around 0.031 microseconds, which still allows the proposed LI framework to effectively work in real-time scenarios. The truth is that even though the proof-of-concept uses exclusive containers, it is executed on a single workstation, which may have affected and caused the aforementioned slight delay.

In conclusion, it is important to emphasize that the deployment of the proposed LI framework has no significant impact on the user QoS in both end-to-end file exchange and real-time VoIP call use cases.

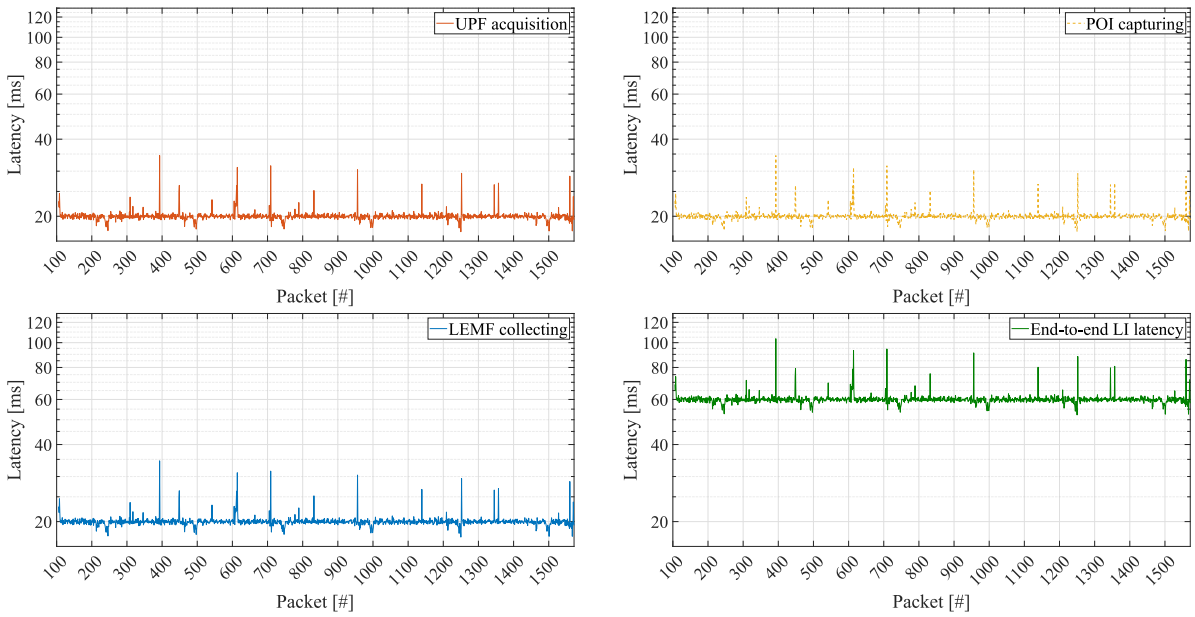


Fig. 6. Packet latency across the different LI stages of a 30 s VoIP call.

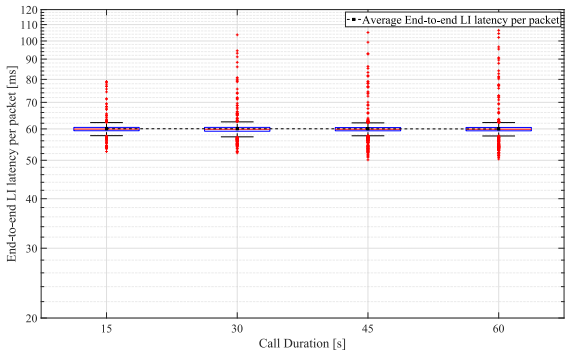


Fig. 7. Statistics of the End-to-end LI latency phase per packet for the four different VoIP call durations.

Table 2
Average delay difference experienced by each packet by deploying or not the proposed LI framework.

	Call Duration [s]	Average delay difference experienced by a single packet [ms]
VoIP call	15	0.077452
	30	0.038487
	45	0.033085
	60	0.049420
	File Size [KB]	**Average delay difference experienced by a single packet [ms]**
File exchange	10	0.000012
	10^2	0.000017
	10^3	0.000031
	10^4	0.000035

5.4. Comparison: VoIP vs. file exchange

In terms of packet latency, both scenarios demonstrate the LI framework's capability to achieve real-time processing. In real-time VoIP calls, the End-to-end LI Latency per packet is consistently around 60 ms, allowing for efficient interception even in short-duration VoIP calls. On the other hand, file exchange interception exhibits a slightly lower latency, with an average End-to-end LI Latency per packet of around 0.25 ms. This implies that the LI framework can handle real-time interception for both scenarios, with a more granular efficiency observed in file exchanges. This difference arises because, when implementing the VoIP call, all cryptographic operations are performed at the same time as sending each SRTP packet, resulting in a higher packet End-to-end LI Latency.

Secondly, the impact of different parameters is noteworthy. In real-time VoIP calls, the call duration does not significantly affect the End-to-end LI Latency, maintaining a constant range across various VoIP call durations. In contrast, in file exchange, the file size has a more pronounced effect on the End-to-end LI latency, with larger files leading to increased average latency, suggesting that the LI framework performance in file exchange is more influenced by the processed packet data size.

Thirdly, by evaluating the packet End-to-End user Latency of both a 30 s VoIP call and a 10^3 KB file exchange, it is evident that the packet size influences the user-experienced latency when the LI framework is active. It is not true, however, that the suggested LI framework appears to have a more significant influence on end-to-end file exchange performances. Indeed, Table 2 displays the average delay difference experienced by each packet by deploying or not deploying the proposed LI framework. Here, it is evident that, in the case of VoIP conversations, the difference is three orders of magnitude more than that of the file exchange scenario.

Lastly, it is noteworthy to highlight that for both real-time VoIP call and end-to-end file exchange scenarios, the impact on the QoS experienced by the user introduced by deploying the proposed LI framework is negligible.

In summary, the scalability of the proposed LI framework becomes apparent in both scenarios. The ability to handle diverse scenarios with minimal impact on latency performance underscores the robustness and adaptability of the proposed LI framework, making it a promising solution for multiple interception applications, ranging from VoIP calls to end-to-end file exchanges.

6. Conclusion

This paper presented a novel LI framework, built on top of the existing 3GPP LI standard, in the context of the evolving 5G and Beyond 5G networks. The proposed LI architecture, based on industry standards,

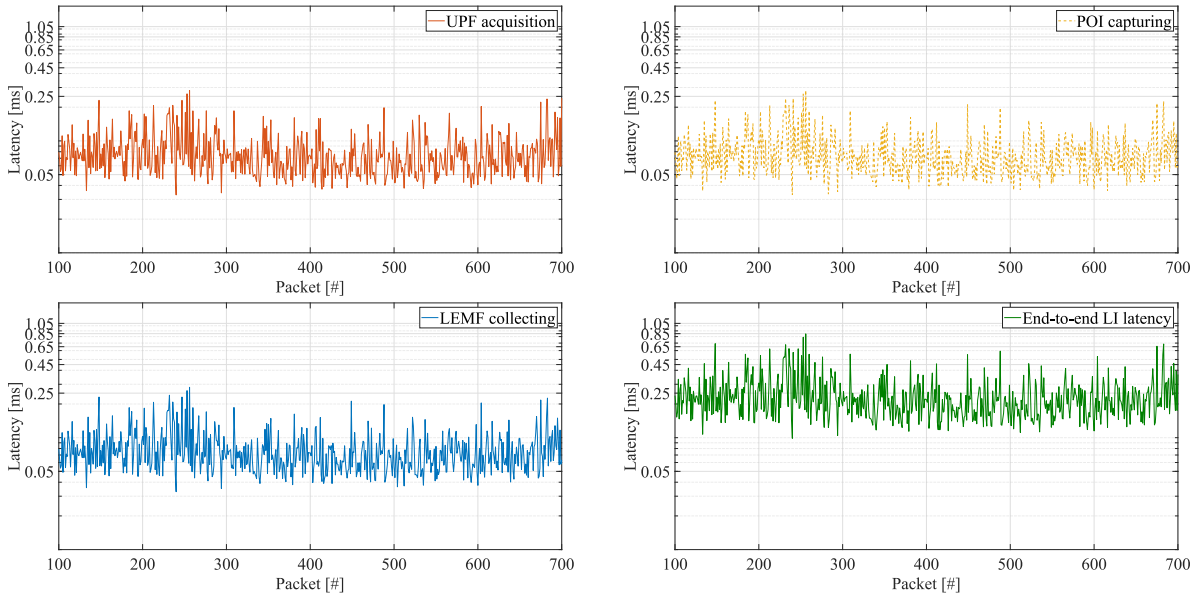


Fig. 8. Packet latency across the different LI stages of a 10^3 KB exchanged file.

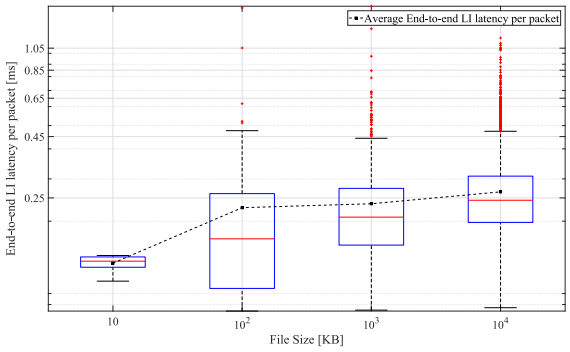


Fig. 9. Statistics of the End-to-end LI latency phase per packet as a function of the four file sizes.

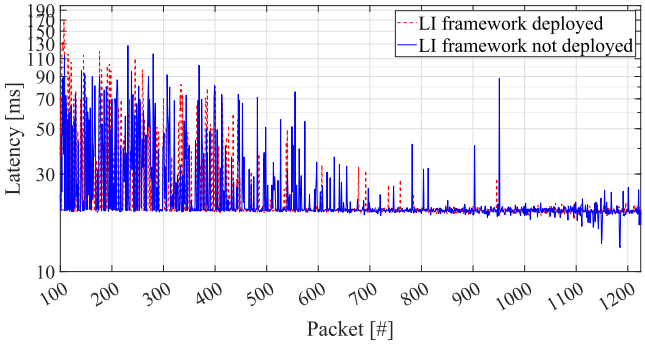


Fig. 10. Packet End-to-End Latency of a 30 s VoIP call.

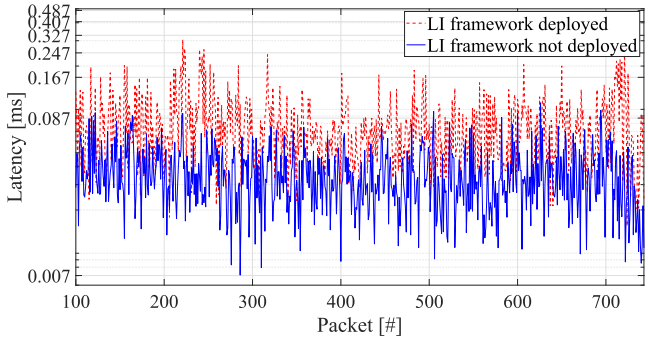


Fig. 11. Packet End-to-End Latency of a 10^3 KB exchanged file.

incorporates Key Escrow capabilities at the application layer. This approach effectively addresses a significant difficulty posed by end-to-end encrypted communications. The implemented proof-of-concept of the conceived LI framework, by emulating key aspects of the 5G network and the LI system, has demonstrated the feasibility of our approach. The experimental tests have further validated the architecture, showcasing its real-time-like latency and scalability potentials, fostering new conversations on standardization and academic bodies. Future studies will focus on (i) the application of this technology within multiple network slices, (ii) the employment of interception procedures at the edge of the network, (iii) the integration of complementary Lawful Interception technologies, and (iv) the possibility of implementing 5G services and LI services on different workstations.

CRediT authorship contribution statement

Ingrid Huso: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Conceptualization. **Marco Olivieri:** Writing – original draft, Software, Data curation, Conceptualization. **Leonardo Galgano:** Writing – original draft, Software, Data curation, Conceptualization. **Adnan Rashid:** Writing – review & editing. **Giuseppe Piro:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Gennaro Boggia:** Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, in the context of partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART", CUP: D93C22000910001), national center on "Sustainable Mobility" (CN00000023 - program "MOST", CUP: D93C22000410001), and partnership on "Cybersecurity" (PE00000007 - program "SERICS", CUP: D33C22001300002, project ISP5G+). It was also supported by the PRIN 2022 projects INSPIRE (grant no. 2022BEXMXN_01) and HORUS (grant no. 2022P44 KA8) funded by the Italian MUR, by the HORIZON MSCA project BRIDGITISE (grant no. 101119554) and the HORIZON JU SNS project 6G-GOALS (grant no. 101139232), and by "The house of emerging technologies of Matera (CTEMT)" project funded by the Italian MIMIT.

Appendix A. Key Escrow Algorithm

This Section aims at technically presenting an in-depth description of the used *IDBC Key Escrow Algorithm* designed and developed in [29]. Specifically, let us suppose that UE A is the under surveillance subscriber and that the LEA presents, via the interface HI1, the LI warrant for intercepting a specific communication session between UE A and UE B to the AUSF. Thus, the ***Key Negotiation Phase*** of the LI framework is detailed below.

Here, the TKA possesses a master secret key $M \in \mathcal{Z}_p^*$ and computes public/private key pairs for subscribers based on their unique identities. Assuming that UE A and UE B share their unique identities, ID_A and ID_B with the TKA, it employs a hash function $\mathcal{H} : \mathcal{Z}_p^* \rightarrow \mathcal{G}$ to generate:

$$\begin{cases} p_A = \mathcal{H}(ID_A) & \text{UE A public key,} \\ P_A = M\mathcal{H}(ID_A) & \text{UE A private key,} \end{cases}$$

and

$$\begin{cases} p_B = \mathcal{H}(ID_B) & \text{UE B public key,} \\ P_B = M\mathcal{H}(ID_B) & \text{UE B private key,} \end{cases}$$

where, how specified into [29], retrieving M is computationally complex as solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) has proved in [36].

Furthermore, as the TKA functions as the 'Escrow Agency,' there is no need for an additional Key Escrow process.

Additionally, the initial sharing of the key k_A between AUSF and UE A, as well as the key k_B between AUSF and UE B, is established.

Firstly, UE A sends μ_1 to the AUSF using the equation:

$$\mu_1 = e_{k_A}(ID_A \| ID_B \| r_A \parallel sign_A(r_A)),$$

where $\|$ denotes concatenation, e_{k_A} denotes the encryption function adopting the shared key k_A, r_A is a random integer generated by the UE A, and $sign_A(r_A)$ is its corresponding signature.

Thus, the AUSF receives and decrypts μ_1, verifies r_A with the signature $sign_A(r_A)$, and then constructs μ_2 to be sent to UE B using the equation:

$$\mu_2 = e_{k_B}(ID_A \| ID_B \| r_A \parallel \text{sign}_A(r_A)),$$

here e_{k_B} denotes the encryption function using the shared key k_B, r_A is the random integer generated by the UE A, and $sign_A(r_A)$ is its corresponding signature.

Upon receiving μ_2, UE B first decrypts and verifies r_A with the signature $sign_A(r_A)$. Subsequently, UE B forwards μ_3 to the AUSF using the equation:

$$\mu_3 = e_{k_B}(ID_B \| ID_A \| r_B \parallel \text{sign}_B(r_B)),$$

Here, e_{k_B} represents the encryption function using the shared key k_B, r_B is a randomly generated nonce by UE B, and $sign_B(r_B)$ is the corresponding signature.

Moreover, UE B is able now to compute $\eta = devf(r_A, r_B)$, where $devf$ is a derivation function from r_A and r_B.

Upon verifying μ_3, the AUSF firstly calculates $\eta = devf(r_A, r_B)$ and then generates μ_4 and sends to the UE A:

$$\mu_4 = e_{k_A}(ID_B \parallel ID_A \parallel r_B \parallel \text{sign}_B(r_B)),$$

where e_{k_B} denotes the encryption function using the shared key k_B, r_B is a randomly generated nonce by UE B, and $sign_B(r_B)$ is the corresponding signature.

Thus, the UE A is now able to verify the identity of UE B and then compute $\eta = devf(r_A, r_B)$.

Once the communication procedure among the subscribers is completed, the interception procedures continue with the AUSF sending τ_1 (containing the LEA request) to the TKA:

$$\tau_1 = \eta \| ID_A \| LEA\ request\ .$$

Subsequently, the TKA sends τ_2 to the LEA via the HI2 interface:

$$\tau_2 = \eta \cdot M\mathcal{H}(ID_A),$$

where the "·" operator denotes the multiplication.

At this point, all entities possess the necessary cryptographic material to generate the communication session key k_{AB} through which the end-to-end communication session will be encrypted.

Firstly, UE A computes $k_{AB} = e(\eta \cdot M\mathcal{H}(ID_A), \mathcal{H}(ID_B))$. Secondly, UE B computes $k_{BA} = e(\mathcal{H}(ID_A), \eta \cdot M\mathcal{H}(ID_B))$.

Here, the function e(·) defines the bilinear function operation, and using IDBC-based model properties described in [29], the validity of the two equations is proven as follows:

$$\begin{aligned} k_{AB} &= e(\eta \cdot M\mathcal{H}(ID_A), \mathcal{H}(ID_B)) = \\ &= e(\mathcal{H}(ID_A), \mathcal{H}(ID_B))^{\eta \cdot M} = \\ &= e(\mathcal{H}(ID_A), \eta \cdot M\mathcal{H}(ID_B)) = k_{BA}. \end{aligned}$$

While the LEA employs a public hash function $\mathcal{H} : \mathcal{Z} \rightarrow \mathcal{P}$ to calculates $\mathcal{H}(ID_A)$ and $\mathcal{H}(ID_B)$, it uses τ_2 to compute the communication session key k_{AB}.

References

[1] The European Commission, Eurostat, Recorded Offences by Offence Category - Police Data, The European Commission, 2023.
[2] The European Commission, Eurostat, Recorded Offences by Offence Category - Police Data, The European Commission, 2021.
[3] C.-X. Wang, X. You, X. Gao, X. Zhu, Z. Li, C. Zhang, H. Wang, Y. Huang, Y. Chen, H. Haas, J.S. Thompson, E.G. Larsson, M.D. Renzo, W. Tong, P. Zhu, X. Shen, H.V. Poor, L. Hanzo, On the road to 6G: Visions, requirements, key technologies, and testbeds, IEEE Commun. Surv. Tutor. 25 (2) (2023) 905–974.
[4] Council of the European Union and EUROPOL, Position Paper on 5G, Tech. Rep., The European Commission, 2019.
[5] I. Palamà, F. Gringoli, G. Bianchi, N. Blefari-Melazzi, IMSI catchers in the wild: A real world 4G/5G assessment, Comput. Netw. 194 (2021) 108137.
[6] 3GPP, 3GPP Release 15 Description, Technical Report (TS) 33.126, 3rd Generation Partnership Project (3GPP), 2022, Release 15.
[7] M. Alatawi, N. Saxena, SoK: An analysis of end-to-end encryption and authentication ceremonies in secure messaging systems, in: Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 187–201.
[8] T. Isobe, R. Ito, Security analysis of end-to-end encryption for zoom meetings, IEEE Access 9 (2021) 90677–90689.
[9] Y. Li, Y. Yu, W. Susilo, Z. Hong, M. Guizani, Security and privacy for edge intelligence in 5G and beyond networks: Challenges and solutions, IEEE Wirel. Commun. 28 (2021) 63–69.
[10] M. Vidoni, E. Senior Course, F. Police, 5G technology: New challenges for law enforcement agencies to face, Eur. Law Enforcement Res. Bull. 22 (2022) 157–171.
[11] Scientists4Crypto, Academic letter to the European commission on "encryption — Security through encryption and security despite encryption", Scientists4Crypto (2020).
[12] M. Säily, O.N.C. Yilmaz, D.S. Michalopoulos, E. Pérez, R. Keating, J. Schaepperle, Positioning technology trends and solutions toward 6G, in: 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, 2021.

[13] D. Giustiniano, G. Bianchi, A. Conti, S. Bartoletti, N.B. Melazzi, 5G and beyond for contact tracing, IEEE Commun. Mag. 59 (9) (2021) 36–41.

[14] V. Doronin, "Lawful interception – A market access barrier in the European union"? Computer Law & Security Review 51 (2023) 105867.

[15] M. Monshizadeh, V. Khatri, M. Varfan, R. Kantola, LiaaS: Lawful interception as a service, in: 2018 26th International Conference on Software, Telecommunications and Computer Networks, SoftCOM, 2018.

[16] G. Ungaro, F. Ricchitelli, I. Huso, G. Piro, G. Boggia, Design and implementation of a lawful interception architecture for B5G systems based on key escrow, in: 2022 IEEE Conference on Standards for Communications and Networking (CSCN), CSCN'22, Thessaloniki, Greece, 2022.

[17] 3GPP, Lawful Interception (LI) Requirements, echnical Report (TS) 33.126, 3rd Generation Partnership Project (3GPP), 2022, Release 18.0.0.

[18] 3GPP, Lawful Interception (LI) Architecture and Functions, Technical Report (TS) 33.127, 3rd Generation Partnership Project (3GPP), 2023, Release 18.5.0.

[19] 3GPP, Protocol and Procedures for Lawful Interception (LI), Technical Report (TS) 33.128, 3rd Generation Partnership Project (3GPP), 2023, Release 18.5.0.

[20] N. Borisov, I. Goldberg, E. Brewer, Off-the-record communication, or, why not to use PGP, in: Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society, WPES '04, Association for Computing Machinery, New York, NY, USA, 2004, pp. 77–84.

[21] R. Stedman, K. Yoshida, I. Goldberg, A user study of off-the-record messaging, in: Proceedings of the 4th Symposium on Usable Privacy and Security, SOUPS '08, Association for Computing Machinery, New York, NY, USA, 2008, pp. 95–104.

[22] K. Cohn-Gordon, C. Cremers, L. Garratt, J. Millican, K. Milner, On ends-to-ends encryption: Asynchronous group messaging with strong security guarantees, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 1802–1819.

[23] WhatsApp Messenger, Whatsapp Encryption Overview, Tech. Rep., 2023, URL https://scontent-ams4-1.xx.fbcdn.net/v/t39.8562-6/383236184_722587863039320_5040651063228680393_n.pdf?_nc_cat=101&ccb=1-7&_nc_sid=b8d81d&_nc_ohc=2mCqgHDwvkkAX-SZwA-&_nc_ht=scontent-ams4-1.xx&oh=00_AfDx5tswN8agfAV4lVpUamJBsr21BnOqtbIch35wpRE8RQ&oe=659961C4. (Accessed 1 December 2023).

[24] Telegram Messenger LLP, Telegram API Documentation, Tech. Rep., 2023, URL https://core.telegram.org/api/end-to-end. (Accessed 1 December 2023).

[25] T.-H. Kim, W.-B. Kim, D. Seo, I.-Y. Lee, A secure encapsulation schemes based on key recovery system, in: Silicon Valley Cybersecurity Conference: First Conference, SVCC 2020, San Jose, CA, USA, December 17–19, 2020, Revised Selected Papers 1, Springer, 2021, pp. 25–37.

[26] H. Farlow, B.M. Edwards, Shining a light on 'going dark': A framework to guide the co-design and communication of decryption laws based on the passage of the telecommunications and other legislation (assistance and access) bill 2018, Comput. Law Secur. Rev. 46 (2022) 105726.

[27] T. Riebe, P. Kühn, P. Imperatori, C. Reuter, US security policy: The dual-use regulation of cryptography and its effects on surveillance, Eur. J. Secur. Res. 7 (1) (2022) 39–65.

[28] C. Duan, J. Grimmelmann, Content moderation on end-to-end encrypted systems: A legal analysis, 2023, Available at SSRN 4457414.

[29] K. Han, C.Y. Yeun, T. Shon, J. Park, K. Kim, A scalable and efficient key escrow model for lawful interception of IDBC-based secure communication, Int. J. Commun. Syst. 24 (4) (2011).

[30] T. Kim, W. Kim, D. Seo, I. Lee, Secure encapsulation schemes using key recovery system in IoMT environments, Sensors 21 (10) (2021).

[31] K. Han, C.Y. Yeun, K. Kim, New key escrow model for the lawful interception in 3GPP, in: 2009 Digest of Technical Papers International Conference on Consumer Electronics, 2009, pp. 1–2.

[32] A. Di Felice, Encryption: Finding the Balance Between Privacy, Security and Lawful Data Access, Position Paper on Encryption Policy, DIGITALEUROPE, 2020, URL https://cdn.digitaleurope.org/uploads/2020/03/DIGITALEUROPE-Position-on-Encryption-Policy-.pdf.

[33] A.K. Ranjan, V. Kumar, M. Hussain, Security analysis of TLS authentication, in: 2014 International Conference on Contemporary Computing and Informatics, IC3I, IEEE, 2014, pp. 1356–1360.

[34] A. Ferreira, R. Giustolisi, J.-L. Huynen, V. Koenig, G. Lenzini, Studies in socio-technical security analysis: Authentication of identities with TLS certificates, in: 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2013, pp. 1553–1558.

[35] J. Zhang, L. Yang, W. Cao, Q. Wang, Formal analysis of 5G EAP-TLS authentication protocol using proverif, IEEE Access 8 (2020) 23674–23688.

[36] I. Blake, G. Seroussi, N. Smart, Elliptic curves in cryptography, in: London Mathematical Society Lecture Note Series, Cambridge University Press, 1999.

Ingrid Huso received her Master Degree (with honors) in Telecommunication Engineering from Politecnico di Bari, Bari, Italy in April 2021. Her Research interests include Network Security, Privacy Enhancing Technologies (PET), Physical Layer Security, and Internet of Things. Since 2021, she has been a Ph.D student at the Department of Electrical and Information Engineering at Politecnico di Bari.

Marco Olivieri received his Bachelor's degree (with honors) in Electronic and Telecommunications Engineering in 2022 from Politecnico di Bari. Concurrently, he is pursuing his Master's degree in Telecommunications Engineering. His research interests encompass Network Security, mobile and non-terrestrial networks. Since February 2023 he cooperates in research activities within the Telematics Lab of Politecnico di Bari.

Leonardo Galgano received his Master's degree (with Hons.) in Telecommunication Engineering with a focus on cybersecurity in 2023 from Politecnico di Bari. His research interests include Network Security and in particular lawful interception in beyond 5G networks.

Adnan Rashid, a member of IEEE and the Internet Society, currently serves as an Assistant Professor at Politecnico di Bari, Italy, since March 2023. He earned his Ph.D. in Telecommunication and Telematics from the University of Florence in 2022 and holds an MS in Computer Engineering from CUST, Pakistan (2015). His research focus is on IoT system security, protocols, and applications. He is actively engaged in the IETF working group (ippm), and he contributes to IPv6 Performance and Diagnostic Metrics Version 2. He is a dedicated Technical Program Committee member for prestigious conferences and collaborates on the IETF 6LoWPAN-ND protocol's development with ns-3 simulator maintainers.

Giuseppe Piro is an Associate Professor at "Politecnico di Bari", Italy. He received the Ph.D. degree in Electronic Engineering, a first, and a second level degree (both with Hons.) in Telecommunications Engineering from "Politecnico di Bari", Italy, in 2006 and 2008, respectively. His main research interests include wireless networks, network simulation tools, 5G and beyond, network security, nano-scale communications, Internet of Things, and Software-Defined Networking. He serves as Associate Editor for Internet Technology Letter (Wiley), Wireless Communications and Mobile Computing (Hindawi), and Sensors (MDPI).

Gennaro Boggia received the Dr.Eng. and Ph.D. degrees (with Hons.) in electronics engineering from the Politecnico di Bari, Bari, Italy, in July 1997 and March 2001, respectively. He is a Full Professor of Telecommunication with the Politecnico di Bari, Bari, Italy. His research interests include wireless networking, cellular communication, protocol stacks for industrial applications and smart grids, Internet measurements, and network performance evaluation.