O-RAN next Generation Research Group (nGRG)

Contributed Research Report

Research Report on Data Privacy-Preserving Techniques for O-RAN

Report ID: RS04-2025-RI03

Contributors:

Rakuten Symphony

Nokia

Qualcomm

Rakuten Mobile

Release date: 2026.06.01

Authors

Company	Name	Email
Rakuten Symphony	Krishna Pramod A Paromita Shah	krishna.adharapurapu@rakuten.com paromita.chintanshah@rakuten.com
Nokia	Emad Heydari Beni Rakshesh P Bhatt Clifton Fernandes	emad.heydari_beni@nokia-bell-labs.com rakshesh.p_bhatt@nokia.com clifton.fernandes@nokia.com
Qualcomm	Soo Bum Lee	soobuml@qti.qualcomm.com
Rakuten Mobile	Antoinette Ngye	antoinette.ngye@rakuten.com

Reviewers

Company	Name	Email
Reliance Jio	Vikas Dixit	Vikas1.Dixit@ril.com
1FINITY	Nishant Tiwari	Nishant.Tiwari@fujitsu.com
Qualcomm	Mike Garyantes	mgaryant@qti.qualcomm.com

Disclaimer

The content of this document reflects the view of the authors listed above. It does not reflect the views of the O-RAN ALLIANCE as a community. The materials and information included in this document have been prepared or assembled by the above-mentioned authors and are intended for informational purposes only. The above-mentioned authors shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of this document subject to any liability which is mandatory due to applicable law. The information in this document is provided ‘as is,' and no guarantee or warranty is given that the information is fit for any particular purpose.

Copyright

The content of this document is provided by the above-mentioned authors. Copying or incorporation into any other work, in part or in full of the document in any form without the prior written permission of the authors is prohibited.

Executive summary

This document explores privacy-preserving techniques and their recent advancements, specifically focusing on their application to data within O-RAN network functions. The disaggregation of the telecom network, a key characteristic of O-RAN, introduces the imminent possibility of data being shared across multi-vendor O-RAN assets for various use cases, thereby necessitating robust data privacy measures.

The report investigates how data privacy-preserving techniques influence AI/ML models during both training and inference phases and their impact on the overall performance of network functions. Particular attention is given to the data shared from rApps and xApps on Non-RT RIC and Near-RT RIC platforms, especially over the A1, Y1, E2 interfaces, and for the usage of AI/ML. Beyond privacy, the trustworthiness of the data and models employed during AI/ML training and inference is also a critical security consideration.

Furthermore, this report addresses the aspects of verifying and auditing data privacy preservation, emphasizing its importance for regulatory and legal compliance. This report explores how maturing privacy-preserving techniques like homomorphic encryption, federated learning, differential privacy, multi-party computation and unified data representation are chosen based on integrity, confidentiality, and privacy needs for O-RAN in future 6G networks, with an eye towards standardization.

Table of Contents

List of abbreviations

A1	O-RAN interface between Near-RT RIC and Non-RT RIC
A1-EI	A1 enrichment information service
BCRs	Binding Corporate Rules
BGW	Ben-Or-Goldwasser-Wigderson
BMR	Beaver-Micali-Rogaway
CAM	Cooperative Awareness Message
CCPA	California Consumer Privacy Act
CN-RAN	Converged Radio Access Network
E2	O-RAN interface between Near-RT RIC and E2 Node
E2 Node	One of O-CU-CP, O-CU-UP, O-DU or O-eNB
FHE	Fully Homomorphic Encryption
FL	Federated Learning
GDPR	General Data Protection Regulation
GMW	Goldreich-Micali-Wigderson
HE	Homomorphic Encryption
KPI	Key Performance Indicator
ML	Machine Learning
MPC	Multiparty Computation
Near-RT RIC	Near Real Time RAN Intelligent Controller
Non-RT RIC	Non-Real Time RAN Intelligent Controller
NF	Network Function
O-CU	O-RAN Central Unit
O-CU-CP	O-RAN Central Unit (Control Plane)
O-CU-UP	O-RAN Central Unit (User Plane)
O-DU	O-RAN Distributed Unit
O-RU	O-RAN Radio Unit
OT	Oblivious Transfer
PHE	Partially Homomorphic Encryption
PII	Personally Identifiable Information
PPT	Privacy-Preserving technique
PM	Performance Measurements
RAI	RAN Analytics information
rApp	Non-RT RIC Application
RIC	RAN Intelligent Controller
SCCs	Standard Contractual Clauses
SHE	Somewhat Homomorphic Encryption
SMO	Service Management and Orchestration
SMPC	Secure Multi-Party Computation
WG1	Work group 1 (Use Cases and Overall Architecture Workgroup)
WG11	Work group 11 (Security work group)
xApp	Near-RT RIC Application
ZTA	Zero trust Architecture
ZTMM	Zero trust maturity model

List of figures

1 Introduction

This research report explores privacy-preserving techniques and recent advances in the industry, examining their application to data handled by O-RAN architecture elements. It analyzes the impact of these techniques on AI/ML models during training and inference stages, as well as their effect on the performance of architecture elements. Additionally, this report discusses methods for verifying and auditing data privacy preservation, which is essential from both a regulatory and legal standpoint.

This report evaluates the applicability and effectiveness of various privacy-preserving techniques, including homomorphic encryption, differential privacy, federated learning, secure multi-party computation, and unified data representation for anonymization, within the O-RAN context. It highlights that the successful integration of these techniques can address privacy concerns in disaggregated RAN environments.

1.1 Background

The O-RAN architecture enables data collected by various O-RAN architectural elements to be utilized for various algorithms and ML model training/inference. This data may include sensitive non-personal data [40] (such as network metrics, location patterns, or device-specific performance data) that should not be accessible to other Network functions, services, or elements in its clear (unencrypted) form.

Currently, data security requirements are defined for different types of data and scenarios where data is transmitted, stored, or used, as specified by the O-RAN ALLIANCE WG11 (security work group). WG11's work includes foundational privacy-preserving requirements, however, detailed privacy use cases and a comprehensive exploration of data privacy aspects are areas where further research is beneficial. This report aims to comprehensively explore the data privacy aspects of O-RAN and the application of privacy-preserving techniques to this data.

1.2 Objectives and Scope

The primary objective of this research report is to survey and analyze different techniques that could be used in O-RAN for data privacy protection, with a focus on future O-RAN networks. The study will evaluate applicability and effectiveness of privacy-preserving techniques for non-personal data (including anonymized data [40]).

It is important to note that the Open Radio Access Network (O-RAN) does not typically store and process subscriber information directly which is PII by default, as the management of such data is primarily handled by the Core Network (e.g. 5G core network as defined in 3GPP TS 23.501 [73]). Hence, in this report, the privacy aspects of use cases handling non-personal data belonging to different O-RAN network functions are considered.

Following Use Cases Considered:

- Data shared with third-party applications (e.g., xApps and rApps) by the RIC platform:

    - Data shared over the E2 interface
    - Data shared over the Y1 interface
- Enrichment data shared over the A1 interface between Non-RT RIC and Near-RT RIC

The following privacy-preserving techniques are being studied for the above use cases:

- Homomorphic encryption – enabling computations on encrypted data
- Federated learning – allowing model training without raw data exchange
- Differential privacy – ensuring individual data points remain indistinguishable
- Secure multi-party computation – enabling joint computations while keeping inputs private
- Unified data representation for anonymization – for example, representation of data as images and videos

2 O-RAN Data Privacy Challenges

The Open Radio Access Network (O-RAN) is a transformative approach to building and operating radio access networks, promoting openness, interoperability, and vendor diversity to enhance flexibility and efficiency. However, this open and disaggregated architecture introduces data privacy challenges.

The involvement of multiple stakeholders, including network operators, equipment vendors, and third-party application providers, further complicates the landscape because each entity often operates with its own distinct security practices and access control policies. This divergence can create potential vulnerabilities, as inconsistent security postures across the ecosystem can lead to gaps, misconfigurations, or unmanaged interfaces that malicious actors can exploit.

Data privacy in O-RAN is critical because the network handles vast amounts of sensitive information, including user data, control data, and operational data. Protecting this data from unauthorized access, interception, and misuse is essential for maintaining user trust and complying with regulatory requirements.

Additionally, the integration of rApps (Non-RT RIC applications) and xApps (Near-RT RIC applications) within the O-RAN ecosystem, along with the use of AI/ML algorithms [12], introduces further data privacy considerations. Specifically, potential threats include data leakages resulting from improper isolation between network elements and unprotected data, as well as malicious xApps and rApps that may exfiltrate or interfere with private data and inference attacks on AI/ML models used in RIC functions.

These applications leverage vast amounts of data to optimize network performance and enhance user experiences but can include sensitive user information and operational metrics. Ensuring responsible data handling with robust encryption, anonymization, and access control measures, along with transparency in AI/ML data

processing, is crucial to mitigate privacy risks and build trust among users and stakeholders.

2.1 Challenges and Threats to O-RAN Data Privacy

Following are a few challenges to data privacy in O-RAN:

Increased Attack Surface: The disaggregated nature of O-RAN means more interfaces and components, each of which can be a potential entry point for attackers.

Data Interception: Data transmitted over various O-RAN interfaces can be intercepted if not properly encrypted, leading to unauthorized access and potential data breaches. Insider attacks should also be considered as a potential threat for private data interception and exfiltration.

Third-Party Risks: Involvement of third-party vendors and applications (owing to the openness of O-RAN) increases the risk of private data leakage and misuse.

Compliance and Regulatory Issues: Ensuring compliance with data protection regulations (e.g., GDPR – General Data Protection Regulation) across different jurisdictions can be complex.

Data Obfuscation Complexity: Concealing or scrambling privacy-sensitive data before providing it as an input (e.g. to RIC) can be an additional challenge for such a variety of data, with huge volumes.

Lack of Protection: Not using any privacy protection measures for sensitive data poses a security threat.

Performance Overhead: Privacy protection measures can add pre-processing time, and further increase computing resource requirements.

Automation Imperative: Considering the volumes of data, automating every step of data analytics is also a mandate, including privacy protection measures.

From a risk perspective, privacy risks from third-party application (xApp / rApp) data handling and data exfiltration risks stemming from the ML model supply chain (e.g., model inversion or leakage of training data) represent some of the most significant threats.

2.2 O-RAN Data Assets and Privacy Protection

O-RAN being at the edge, closer to users/UEs compared to other network elements can collect granular UE specific information such as throughput, packet delay, packet loss etc.

This data can be shared across an interface or via an API with another application and could contain information that the owner of the data wants to keep private due to the data recipient being in a different trust zone.

An example of such private data could be the metrics collected by a shared O-RU for different carriers belonging to different operators or data shared by Near-RT RIC or Non-RT RIC with xApps and rApps respectively.

Privacy protection techniques facilitate open interfaces by enabling useful computations on private information even when shared across trust boundaries.

2.2.1 Data Diversity in O-RAN: Implications for AI/ML

O-RAN ALLIANCE WG1 Use Cases Analysis Report [18] presents high-level descriptions for use-cases which require AI/ML based implementations in RIC. This list is very likely to grow as the technology evolves in future. Each of these use cases may have different data formats or how AI/ML is applied. This makes the implementation more complex and less generic.

Considering the growing number of cells, and even larger number of devices, the rate and volume of generated data is expected to be enormous. The data consumed by AI/ML applications can include Cooperative Awareness Messages (CAMs), navigation data, geo data (position, velocity, direction), Performance Measurement (PM) data, radio measurement data, trace data, troubleshooting logs, KPIs, alarm data, Minimization of Drive Testing (MDT) / Radio Link Failure (RLF) reports, QoE reports, etc. All these various kinds of input data have different formats with varying statistical properties. Moreover, even for the same datasets, requirements change with the use cases.

The pre-processing and logistics to develop AI/ML models differ for individual use cases. For instance, when developing a model to predict KPIs, due to the statistical properties each KPI requires a different learning process to generate a model with good accuracy.

From an O-RAN Near-RT RIC perspective, the data pipelining aspects and the APIs for setting up these pipelines are explicitly defined in [10]. Specifically, section 6.2.9.1 of [10] provides the definition of data pipelining and outlines how datasets, ready to be consumed by AI/ML models, can be prepared using this approach. For example, Annex A of [10] details a pre-processing method to preserve privacy that can be integrated as part of a data pipeline.

For a time-series prediction, for example, it is required to make the input data statistically as stationary as possible to get accurate predictions. In mathematics and statistics, a stationary process is a stochastic process whose statistical properties, such as mean and variance, do not change over time. More formally, the joint probability distribution of the process properties (e.g. mean and variance) remains the same when shifted in time. This implies that the process is statistically consistent across different time periods. Because many statistical procedures in time series analysis assume stationarity, non-stationary data are frequently transformed to achieve stationarity before analysis. The methods used to make a time-series stationary can be different for different KPIs or counters because the nature of their non-stationarity often varies. For example, a KPI with variance that increases with its mean might benefit from a logarithmic transformation, while another exhibiting a linear trend might require differencing. Therefore, the specific characteristics of the data, which are often tied to the KPI type, dictate the most appropriate pre-processing technique (e.g., logarithmic, square-root, or differencing methods) to achieve stationarity.

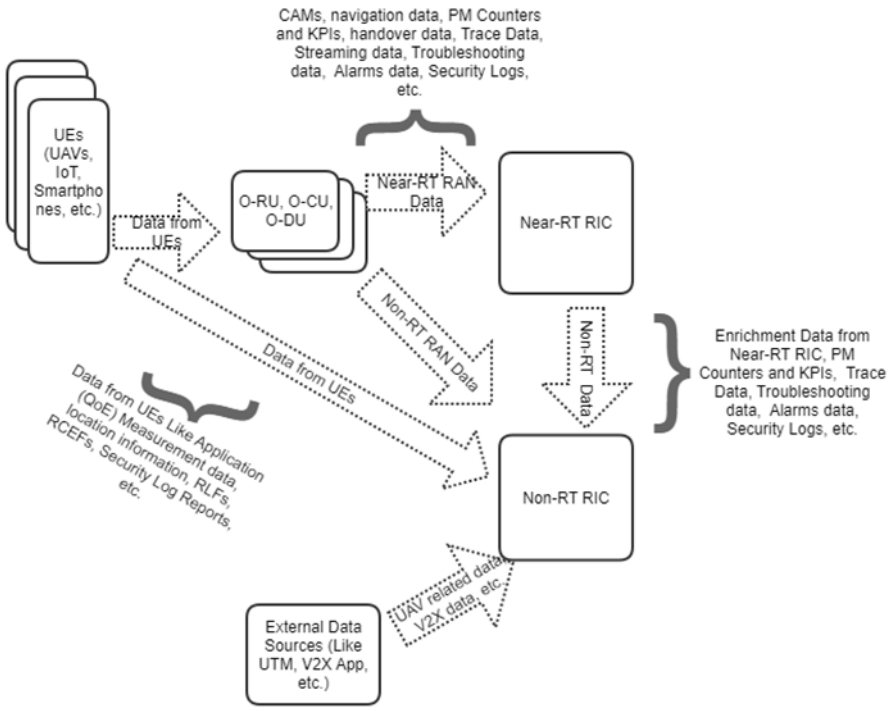


Figure 1: High-level view of input data to Near-RT and Non-RT RIC

2.2.2 Data Shared to Third Party Applications by the RIC platform

The data shared with third-party applications, in some cases, should be protected against unwanted exposure which can result in loss of privacy. For the purpose of this report, O-RAN data assets are categorized as follows:

- **UE-Level Data**: Throughput, packet delay, loss, and mobility patterns.
- **Network-Level Data**: RIC telemetry, spectrum efficiency, and base station load.
- **Service-Level Data**: Application usage trends, Quality of Experience (QoE) metrics, and network slice performance.

This data may be used for analytics by the third-party applications directly or relayed to other applications for processing. Ensuring that the data privacy is maintained across different applications which are processing the data is an issue if the end-to-end path that the data takes is not in the control of the data owner. For example, the following data flows in RIC have the potential for unintended leakages:

- E2 Interface → Near-RT RIC → xApp → External API (potential leakage at xApp level).
- A1 Interface → Non-RT RIC → rApp → Third-party cloud analytics (privacy risk if enrichment data is stored long-term).

To some extent, secure transport of data, authentication and authorization of the data recipient could prevent data privacy issues. But, once the data is not within the boundary controlled by the data owner, it is not always easy to ensure its privacy.

For example, the data could be used as training data for an ML algorithm, the resulting ML model could be transferred to an entity that is not trustworthy from the data owner point of view. The entity receiving the data can then use data exfiltration techniques on the ML model to violate privacy aspects of the shared data.

A significant concern is the potential for re-identification, even from seemingly aggregated or "non-personal" data. For instance, while UE-level KPIs like throughput, packet delay, and mobility patterns may appear anonymous individually, combining these with temporal and geographical information (e.g., cell tower IDs, time of day) can enable the reconstruction of unique user behavior and movement trajectories.

Consider a scenario where anonymized UE-level data is received at Near-RT RIC, including frequent location updates and application usage patterns. Even without explicit identifiers, an insider attacker could cross-reference this data with publicly available information (e.g., social media check-ins, known home/work locations) to identify specific users. The unique combination of an individual's mobility and usage patterns creates a distinctive "digital fingerprint," leading to re-identification. For a broader understanding of various threats targeting the Near-RT RIC, including those related to malicious xApps and unauthorized information access, refer to section 7.4.1.4 of O-RAN Threat modelling and risk assessment technical report [74].

2.2.2.1 Data Shared Over E2 Interface

The E2 interface connects Near-RT RIC to E2 nodes. An E2 node can be separate O-CU-CP, O-CU-UP, O-DU or O-RU NF, or an aggregated node of these NFs, including Near-RT RIC. (See Annex A.4 of [8]).

O-RAN architecture does not limit the deployment scenarios where the aggregated E2 node constitutes NFs from different vendors. This poses an interesting challenge where the aggregated E2 node contains NFs from different vendors and the vendors want to keep certain data private to the NF and not expose it in its raw form over the E2 interface. The O-RAN threat modelling and risk assessment document [74] emphasizes robust isolation, data privacy and security practices for open-source components as key security principles for multi-vendor deployments.

Also, even when the NFs are separate, the NF vendor may want to not expose NF’s private data over E2 interface, since the E2 data could possibly be shared with xApps hosted on the Near-RT RIC.

The Near-RT RIC specifies secure onboarding procedures for xApps [10], and the communication between xApps and Near-RT RIC is protected for secure transport. But the data owner of an E2 node might not have a trust relation with the xApp vendor. Hence, privacy-preserving techniques play a key role in protecting data privacy of E2 data.

2.2.2.2 Data Shared Over Y1 Interface

The Y1 interface is exposed by Near-RT RIC to Y1 consumers. The Y1 interface provides notifications and query APIs to retrieve RAI from Near-RT RIC [6]. Y1 interface is defined to support different types of RAI (e.g., RAN performance analytics which include UE positioning results, including location coordinates, coordinate system, position methods used, achieved location QoS accuracy) [17].

Near-RT RIC does not have prior information of the Y1 consumers and relies on the authorization server for providing the OAuth access tokens with appropriate claims [7]. To avoid issues arising from misconfiguration of claims at the authorization server or misuse/leak of private information by/at the Y1 consumers, the data must be privacy protected.

2.2.3 Enrichment Data Shared Over A1 Interface

The A1-EI service of Non-RT RIC produces and makes the enrichment information (EI) available to Near-RT RIC over the A1 interface. The sources providing the information to produce EI could be within the RAN or outside the RAN (e.g. “Context-based dynamic handover management for V2X” use case in [9]).

The enrichment information may contain sensitive data (e.g. Location information) that needs to be privacy-protected. The privacy considerations for the information stem from the fact that there are different RAN and non-RAN systems and network functions involved with a combination of trust zones as the data is transported from its source to its recipient.

3 Privacy-Preserving Techniques

As networks become increasingly decentralized, more resources are being deployed at the network edge to deliver localized and personalized services. This shift necessitates the development of next-generation systems specifically designed to protect and preserve data privacy [1][11].

This section describes the recent trends in privacy-preserving techniques and examines their applicability to data privacy in O-RAN.

It is important to note that the various privacy-preserving techniques examined here are not mutually exclusive and can be combined to achieve desired outcome for a specific scenario. Given O-RAN’s disaggregated and multi-vendor nature introduces trust issues, making the implementation of privacy techniques critical. Furthermore, data sharing across open interfaces (e.g., E2, A1, Y1) necessitates privacy enforcement mechanisms that go beyond basic security controls.

3.1 Homomorphic Encryption

3.1.1 Concept

Homomorphic encryption (HE) is a cryptographic technique that allows computations to be performed on encrypted data without needing to decrypt it first. This means that data can remain encrypted while being processed, ensuring that sensitive information

is never exposed during computation. HE enables mathematical operations on the ciphertext that, when the result is decrypted, yields the same result as performing those operations on corresponding plaintexts.

The utility of HE within O-RAN is categorized by the extent of operations it supports: Partially homomorphic encryption (PHE), somewhat homomorphic encryption (SHE), and fully homomorphic encryption (FHE) [56].

Fully homomorphic encryption (FHE) is a technique that allows arbitrary computations on encrypted data. However, FHE is computationally intensive and can be inefficient in some scenarios. In the recent past, there have been various advances aimed at making FHE more efficient, often utilizing techniques such as hardware acceleration based on FPGA, ASIC, or GPUs [2].

The key steps of homomorphic encryption are as follows:

1. Key Generation: A public-private key pair is generated. Crucially, additional evaluation keys (sometimes called relinearization or bootstrapping keys) are also generated. These special keys are essential for enabling computations directly on encrypted data without revealing the plaintext.
2. Encryption: Sensitive plaintext data is encrypted using the public key, producing ciphertexts.
3. Homomorphic Evaluation: Instead of decrypting the data, mathematical operations (e.g., addition, multiplication) are performed directly on these ciphertexts. The homomorphic property ensures that the result of these operations on the encrypted data, when later decrypted, is precisely the same as if the operations had been performed on the original, unencrypted plaintext data.
4. Decryption: The resulting ciphertext (from the homomorphic evaluation) is decrypted using the private key to obtain the final plaintext result. This result is the outcome of the computation performed entirely in the encrypted domain.

Figure 2: Homomorphic encryption key steps

The result decryption needs the private key (SK) as shown in Figure 2.

There are different types of homomorphic encryption, each with its own characteristics which are described below.

3.1.1.1 Pre-FHE

This type of HE, known as partially homomorphic encryption (PHE), supports only one type of mathematical operation, such as addition or multiplication, on encrypted data. An example of this is the Paillier cryptosystem [76], which is additively homomorphic. The majority of PHE schemes are based on security assumptions that do not provide

security against quantum computing attacks [75]. Consequently, PHE adoption is no longer encouraged for new systems, unless the PHE is based on post-quantum primitives such as lattices.

Theoretically, any function can be written as a combination of additions and multiplications. After Craig Gentry's breakthrough [16] towards supporting both addition and multiplication in 2009 using lattice-based cryptography, most leading FHE schemes today are based on lattices [2] [13] [56].

3.1.1.2 Somewhat Homomorphic Encryption (SHE)

Somewhat homomorphic encryption (SHE) represents an intermediate class between PHE and FHE, supporting both addition and multiplication operations on encrypted data, but only for a limited number of operations. The primary constraint, as highlighted by [56], is the accumulation of "noise" within ciphertexts. This noise grows with each homomorphic operation slightly during addition and more significantly during multiplication and eventually prevents correct decryption once a threshold is exceeded.

This inherent limitation means SHE can evaluate two types of fundamental operations, often referred to as "gates". In O-RAN, these "gates" can represent elementary steps within privacy-preserving analytics functions (e.g., calculating the average latency using data from multiple vendors with privacy preservation, determining the weighted sum of various encrypted Quality of Service (QoS) metrics for a service slice without exposing raw subscriber traffic patterns) or individual computational layers within machine learning models (e.g., a neuron's weighted sum and activation function performed on encrypted input features), but only for a subset of such computations with constrained depth [56].

SHE schemes are predominantly constructed using lattice-based cryptography, inherently providing post-quantum security.

3.1.1.3 Fully Homomorphic Encryption (FHE)

Fully homomorphic encryption represents the theoretical and practical pinnacle of homomorphic encryption, enabling unlimited arbitrary computations (both additions and multiplications) on encrypted data. FHE is the most robust of the three types of homomorphic encryption, allowing evaluation of arbitrary "circuits" made up of multiple "gate" types of unbounded depth [56]. This capability is achieved through a process called bootstrapping, which "refreshes" ciphertexts to reduce noise accumulation, thereby allowing an unbounded number of operations [56]. Since Gentry's 2009 breakthrough, FHE has evolved rapidly, with modern schemes based on lattice problems ensuring post-quantum security.

For applications requiring long-term security against quantum adversaries, it is crucial to employ HE schemes built upon post-quantum primitives. NIST report [75] provides migration considerations and provides guidance on timeline for transition.

3.1.1.4 Post-Quantum cryptography based homomorphic encryption (PQC-HE)

PQC-HE designates HE schemes resilient to quantum computer attacks, a vital aspect for long-term data confidentiality. While many modern SHE and FHE schemes are

inherently post-quantum due to their lattice-based foundations, the term PQC-HE explicitly highlights this quantum-resistant property, distinguishing them from schemes requiring adaptation to post-quantum primitives.

3.1.1.5 Lattice-Based Homomorphic Encryption Algorithms

Lattice-based cryptography forms the bedrock of most modern homomorphic encryption (HE) schemes, particularly those offering somewhat homomorphic encryption (SHE) and fully homomorphic encryption (FHE) capabilities. This class of HE allows more flexible computation of functions by enabling the execution of both additions and multiplications on encrypted data. All practical HE schemes with robust performance are based on a lattice concept known as Learning with Errors (LWE) [56] or Ring-LWE (RLWE). LWE/RLWE problems include a certain type of noise that mathematically makes these schemes post-quantum secure against known quantum computing attacks [57].

However, in HE schemes, this noise accumulates while computing on ciphertexts. Too much noise can corrupt the plaintext data, rendering decryption impossible. To remedy this problem and enable deeper computations, there is a procedure called Bootstrapping that can decrease the noise and, hence, provide more room for computation. Therefore, there are two primary categories of HE schemes based on their noise management and computational depth:

3.1.1.5.1 Categories of HE schemes

A. Leveled homomorphic encryption (LHE): This category, which encompasses SHE, supports the evaluation of functions with bounded, predetermined depths. The "depth" refers to the maximum number of consecutive multiplications that can be performed before noise accumulation becomes prohibitive [58]. LHE schemes avoid the costly bootstrapping process, making them more efficient for computations that fit within their noise budget.
B. Fully homomorphic encryption (FHE): This category supports the evaluation of functions with unbounded depths. This is achieved through the periodic application of Bootstrapping, which refreshes the ciphertext's noise level, allowing for an arbitrary number of operations [56].

For brevity and common usage, FHE is often used interchangeably with HE in contexts where the discussion implies the full capabilities of homomorphic computation. However, it is crucial to distinguish between LHE (including SHE) and FHE based on their bounded versus unbounded computational capabilities.

3.1.1.5.2 Types of Lattice-Based FHE Schemes by Operation

Lattice-based FHE schemes are broadly categorized by the type of operations they are optimized to perform on encrypted data:

A. Integer Operations: These schemes are designed for integer arithmetic, supporting addition and multiplication of encrypted integer values. The well-known schemes in this category are BGV (Brakerski-Gentry-

Vaikuntanathan) and BFV (Brakerski/Fan-Vercauteren). They are suitable for applications requiring exact numerical precision.

B. Boolean Operations: These schemes support bitwise Boolean operations such as AND, NOT, OR, XOR, and NOR on encrypted binary data. The prominent schemes in this category are FHEW (Fastest Homomorphic Encryption in the West) [59] and TFHE (Fast Fully Homomorphic Encryption over the Torus) [59]. These schemes are known for their particularly fast Bootstrapping, making them efficient for logic-intensive computations.

C. Approximate Operations: These schemes support fixed-point arithmetic (approximating float values) on encrypted data. The well-known scheme in this area is CKKS (Cheon-Kim-Kim-Song) [60]. CKKS is specifically designed for scenarios where exact numerical precision is not of utmost importance, such as AI/ML algorithms or statistical calculations, which often involve real numbers and can tolerate small approximation errors.

3.1.2 Application in O-RAN

In an open shared multi-vendor environment of O-RAN, HE allows the third-party rApps (Non-RT RIC applications) and xApps (Near-RT RIC applications) to process encrypted information without disclosing confidential data.

Employing FHE or LHE where applicable in O-RAN is valuable for privacy-preserving AI/ML tasks such as encrypted network anomaly detection, predictive RAN optimization, secure User Equipment (UE) behaviour analysis. FHE ensures that sensitive information remains private, even when processed by untrusted third-party rApps/xApps. Following are the O-RAN elements and the roles they can play when homomorphic encryption is used for the purpose of ML model training and inference.

3.1.2.1 Lattice-Based LHE (SHE) in O-RAN

Rationale: LHE (encompassing SHE) is highly applicable for O-RAN components requiring bounded computational depth with post-quantum security. Its performance is significantly better than FHE (100-1000x faster for supported operations) because it avoids the costly bootstrapping step [58]. This makes it suitable for many practical AI/ML inference and analytics tasks within O-RAN where the "circuit" depth is known and limited.

Applicable O-RAN Elements/Interfaces:

SMO/Non-RT RIC (Analytics): For privacy-preserving statistical analysis on encrypted multi-vendor data, such as calculating higher-order moments or running simple regression models on encrypted network state data for trend analysis. For analytics with a known, bounded set of operations, meaning the total number of sequential homomorphic computations can be determined and fixed in advance, LHE provides quantum-safe privacy with better performance than FHE [56] [58].

3.1.2.2 Lattice-Based FHE in O-RAN

Rationale: FHE is essential for O-RAN scenarios demanding unlimited computational depth and arbitrary operations with post-quantum security. While computationally more intensive, its ability to support any "circuit" (arbitrary computation) makes it invaluable for complex AI/ML training, iterative optimization, and dynamic data processing.

3.1.2.2.1 Applicable O-RAN Elements/Interfaces:

O-Cloud (Secure Outsourced Computation) can be used for offloading computationally intensive, privacy-preserving tasks (e.g., large-scale data analytics, complex simulations, extensive AI model training) to a public or shared cloud infrastructure without exposing sensitive O-RAN data to the cloud provider. FHE transforms the cloud into a "blind compute engine," allowing it to process encrypted O-RAN data and return encrypted results. This is critical for maintaining confidentiality in multi-tenant cloud environments and leveraging external compute resources without compromising data ownership [62].

In the past decade there have been significant improvements in using HE for practical applications [14] and using libraries optimized for CPUs [15] instead of specialized hardware to perform training and inference on HE data. Given that machine learning (ML) models are heavily based on floating-point numbers, it is preferable to use fully homomorphic encryption (FHE) schemes that are friendly to floating-point data types, such as CKKS, to avoid the need for converting numbers to integers through techniques like quantization.

3.1.2.3 Case study Workflow for application of homomorphic encryption on AI/ML:

The data from the data sources needs to be encrypted using HE to preserve the privacy. Since FHE does not natively support non-linear functions like ReLU or sigmoid (activation layers), AI/ML models designed for use with FHE must be adapted to use polynomial approximations in these activation layers, which may introduce accuracy trade-offs. The model training is adapted to take the HE data as training input and generate a model. Once the model is available, it can be securely transported to an entity performing the inference using the model.

Following are some of the stages that describe the workflow of using homomorphic encryption in O-RAN.

- The O-RAN Network functions that produce and consume data need to classify the data types that need to be privacy preserved.
- The data owners generate HE keys and encrypt the classified data using the public key.
- The homomorphic encrypted data and the keys required for computation (evaluation keys, but not the private key) are transmitted securely over the O-RAN interface towards the Model training entities.
- The AI/ML model training should also have the capability to perform computations directly on encrypted data. This requires adapting the model's

operations to be compatible with the constraints and requirements of homomorphic encryption schemes.

- The model inference entity gets the homomorphic encrypted data and uses the trained model for inference resulting in an output.
- If the output is encrypted, then it must be transmitted back to the entity (data source) holding the private key to perform output decryption and the decrypted data should be transmitted back to the model inference entity for use.

3.1.3 Impact on AI/ML Models

Training AI/ML models on homomorphic encrypted data is computationally intensive and time consuming. Homomorphic encryption schemes, especially fully homomorphic encryption (FHE), require significant computational resources due to the complexity of performing operations on encrypted data [13]. Though recent research in this area shows that major improvements are made in recent times to optimize the computations.

As for handling noise, homomorphic encryption schemes introduce noise into the ciphertext to ensure security. Each homomorphic operation (addition or multiplication) increases the noise level in the ciphertext. It is crucial to manage the noise to maintain the integrity of the encrypted data during computations [16].

Additionally, to reduce computation overhead during inference, we could also use homomorphic encryption in a way such that the model training/inference uses encrypted inputs and produces encrypted outputs, with the final plaintext result obtained via a trusted encryption / decryption proxy.

3.1.4 Verification and Auditing

To ensure that the model training and inference entity using homomorphic encryption (HE) performs the required functionalities, a structured verification process is essential. The following summarizes the key verification steps for each capability:

- Establish a testbed that simulates O-RAN entities, including the RAN, RIC, and User Equipment (UE), with an integrated AI/ML model tailored for the testbed environment.
- Integrate a suitable homomorphic encryption (HE) library and scheme (e.g., OpenFHE, Lattigo, TFHE-rs) to enable efficient encryption, decryption, and homomorphic operations.
- Configure the testbed with an AI/ML model pre-trained on a dataset specifically designed for HE, ensuring alignment with the intended use case.
- Prepare the encrypted data from the RAN/UE at the data source, ensuring it is ready for processing.
- Feed the encrypted data into the AI/ML model, which will perform computations based on the defined test scenario.

- Execute the inference process on the encrypted HE dataset and capture the resulting encrypted output for further analysis.
- Consider employing Succinct Non-interactive Arguments of Knowledge (SNARKs) to facilitate verifiable homomorphic encryption (VFHE), to confirm that computations on encrypted data have been executed correctly.
- Utilize the encrypted output for decryption at the data source, applying it to the application to validate the O-RAN use case scenario effectively.

3.1.5 Performance and Efficiency

HE offers strong privacy guarantees that comes with significant performance costs. The inefficiencies have roots in the inherent complexity of operating on encrypted data. In this subsection, we outline the root-cause of these issues and the existing optimizations.

3.1.5.1 Efficiency challenges

Computational overhead: HE requires complex mathematical arithmetic, including large modular operations on numbers and polynomials, many polynomial multiplications, and ciphertext relinearizations. The order and size of these polynomials can easily get to large numbers [2], and therefore, these operations drastically slow down computations.

Memory overhead: Unlike symmetric cryptography, **HE ciphertexts** are considerably larger than the plaintext versions. This is often measured by a metric known as the *expansion factor*. Encrypted data can grow 100x–1000x larger than plaintext depending on HE parameters for large computations. Furthermore, HE keys, in particular the evaluation keys required by the entity that computes on encrypted data, can be hundreds of megabytes (and sometimes a few gigabytes). This often has a negative impact on performance, and most importantly, it consumes a considerable amount of memory and storage. In the context of O-RAN, these large evaluation keys (often hundreds of MBs) can significantly impact the memory footprint and provisioning requirements of RIC and O-Cloud Platform components that provide storage and processing capabilities.

Bandwidth consumption: due to ciphertext expansion, there is naturally a large bandwidth consumption. Therefore, data transmission between O-RAN components (e.g., Near-RT RIC and xApps/rApps) requires significantly more bandwidth than plaintext processing.

3.1.5.2 Optimizations to improve efficiency

Adequate parameter selection: choosing optimized encryption parameters not only improve security but also can reduce the requirement to run computationally expensive operations like HE bootstrapping.

Batching and SIMD (Single Instruction, Multiple Data): Most FHE schemes offer a feature to pack multiple values into a single ciphertext. This reduces computational redundancy and speeds up the encrypted operations. This feature is popular among

word-wise and approximate schemes such as BGV/BFV and CKKS, but it is typically absent in bit-wise schemes such as TFHE.

Hardware-friendly HE arithmetization (RNS). In HE, we often deal with large numbers because of the large modulus in ciphertexts (e.g., 800 bits). Unfortunately, these numbers do not fit machine words, and if they are left unchanged, we have to incur the overhead of multi-precision integer arithmetic. This often happens in HE since one might choose a large parameter to avoid expensive operations like bootstrapping. To remedy this issue, a residue number system (RNS) allows a large ciphertext to be decomposed into smaller numbers that fit the computer words. If the hardware (e.g. CPU) allows SIMD parallelism, RNS speeds up the computation drastically.

Transciphering schemes: As mentioned earlier, ciphertext expansion affects bandwidth consumption. One way to solve this problem is to use transciphering schemes [77]. Transciphering schemes refer to symmetric encryption schemes that the ciphertexts can be transformed into HE ciphertexts. To be technically more precise, their ciphertexts can be decrypted inside the HE domain. In other words, the data owner in O-RAN encrypts with a scheme like AES with a 1:1 expansion factor and sends the ciphertext across the network. At the computation point, the computing entity can transform the AES ciphertext to HE ciphertext. In this way, we drastically reduce the bandwidth consumption.

Hardware acceleration: Internal HE operations are highly parallelizable (e.g., NTT/FFT-based multiplication). Therefore, employing hardware acceleration techniques is another crucial way to improve performance [78].. The research community as well as the industry have explored and produced solutions to effectively accelerate FHE schemes, leveraging GPUs, FPGA, ASIC, and Photonic processors for operations like FFTs and others [2] [78].

The following table compares Partially (PHE), Somewhat (SHE), and Fully (FHE) Homomorphic Encryption across key parameters, including their computational and memory overheads.

Feature/Parameter	Partially homomorphic encryption (PHE)	Somewhat homomorphic encryption (SHE)	Fully homomorphic encryption (FHE)
Core Capability	Supports one specific type of operation (e.g., addition OR multiplication) any number of times.	Supports a limited number of both addition and multiplication operations.	Supports arbitrary number of both addition and multiplication operations.
Example Schemes	Paillier, ElGamal	BGV, BFV, CKKS (without bootstrapping)	BGV, BFV, CKKS (with bootstrapping), TFHE, FHEW
Key Size (Public/Secret)	Relatively small (e.g., hundreds of bits to a few kilobytes).	Moderate to large (e.g., hundreds of kilobytes to a few megabytes).	Large to very large (e.g., hundreds of megabytes to several gigabytes).
Evaluation Key Size	Not typically applicable/minimal	Moderate (e.g., hundreds of kilobytes to tens of megabytes)	Large to very large (e.g., hundreds of megabytes to several gigabytes)
Ciphertext Size (Expansion Factor)	Small to moderate (e.g., 2× to 10× plaintext size).	Moderate to large (e.g., 10x to 1000× plaintext size).	Large to very large (e.g., 100× to 10000× plaintext size, or more).
Computational Overhead	Low to moderate.	Moderate to high.	Very high.
Memory Overhead	Low.	Moderate.	High (especially for evaluation keys).
Bandwidth Consumption	Low to moderate.	Moderate to high.	High.

"Depth" of Computation	Unlimited for the supported operation.	Limited (determined by noise growth).	Unlimited (through bootstrapping).
Bootstrapping	Not applicable/necessary.	Not applicable/necessary (by definition, SHE doesn't support unlimited operations).	Essential for achieving "full" homomorphic capabilities; computationally very expensive.
Typical Use Cases	Secure voting, private information retrieval, basic aggregation.	Machine learning inference on encrypted data (limited depth), secure comparisons.	Complex machine learning training, advanced privacy-preserving computations, arbitrary programs.
Optimization Focus	Efficient arithmetic for the single supported operation.	Noise management, efficient modular arithmetic, batching.	Efficient bootstrapping, hardware acceleration, RNS, transciphering.
Typical Operation Latency (Time)			
• Encryption/Decryption	µs to low ms	ms to tens of ms	ms to hundreds of ms
• Homomorphic Add	µs to low ms	µs to low ms	µs to low ms
• Homomorphic Multiply	N/A (or very slow if simulated)	ms to tens of ms	ms to hundreds of ms (pre-relinearization)
• Relinearization (after multiplication)	N/A	ms to tens of ms	tens of ms to hundreds of ms
• Bootstrapping	N/A	N/A	hundreds of ms to several seconds

3.2 Federated Learning

3.2.1 Concept

Federated learning (FL) is a distributed machine learning paradigm that allows the training of a shared AI/ML model across multiple entities or nodes without the need to transfer raw data. This ensures that data privacy is maintained, as the data remains on the local entities.

Federated learning is an excellent technique that can be used to contribute input data to an AI/ML model without compromising the data privacy. Combined with edge computing, federated learning is being proposed to train a model based on distributed private data (UE data, Edge node data, etc.) [3].

The initial model is chosen by a central server and distributed to participating entities. After local training, the entities send only the model updates (e.g. model weights), to the central server. The central server then aggregates these updates to refine the shared model and orchestrates the entire federated learning process. This iterative cycle continues, with the improved shared model being sent back to the entities for further local training, until the model achieves the desired performance.

3.2.2 Application in O-RAN

The O-RAN network functions that want to preserve the data privacy would not want to transfer the data outside the trust boundaries since beyond the trust boundaries the privacy of the data is not guaranteed. FL enables the trusted network functions to collaboratively train machine learning models while keeping the data localized on the NF, thus preserving privacy and confidentiality.

FL in O-RAN involves multiple O-RAN components working together to train a machine learning model in a decentralized manner using API endpoints. In this setup,

the Non-RT RIC, SMO, or O-Cloud can serve as the FL central server which aggregates different partial models from the participating entities.

The Near-RT RIC, E2 Nodes including O-CU, O-DU, O-RU can serve as the distributed / participating entities. In order to participate in the federated learning, the Network functions should be equipped with certain capabilities, computational resources (CPU, GPU), memory resources (RAM and storage).

For legacy network functions that are not equipped with the required software or computational resources, a federated learning proxy service could be used to offload the participation of network functions in federated learning.

If an FL proxy service is used, the proxy (hosted within the trust boundaries of a Network Function) participates in the federated learning by training on the provided data set and then shares the model parameters (partial model) with the aggregator (FL central server). Based on the convergence of the model at the FL central server, the proxy may receive the aggregated updates for re-training.

Workflow for application of federated learning:

Figure 3 is a flow diagram of federated learning with the help of a FL proxy service.

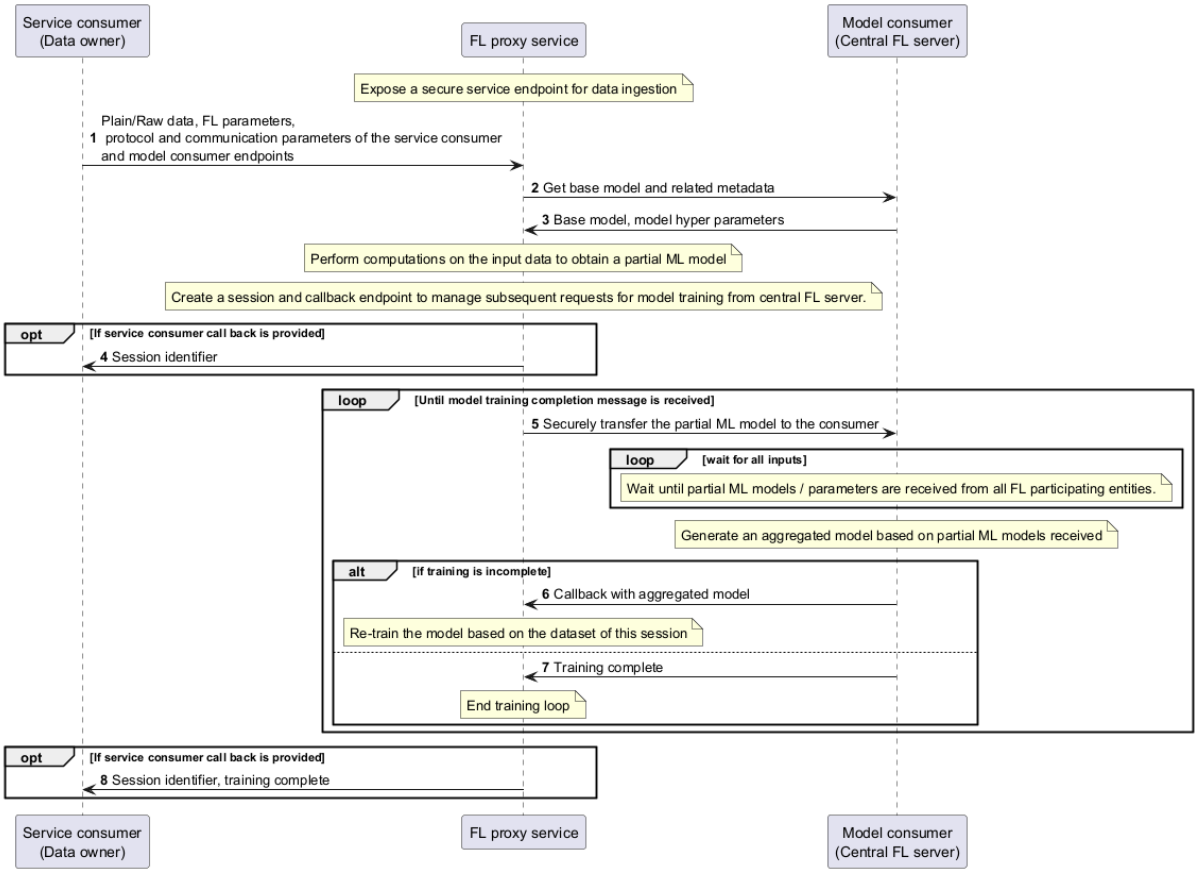


Figure 3: FL proxy service to support legacy/constrained NFs

Below are some of the stages that describe the flow depicted in Figure 3 when using federated learning in O-RAN. Annex B can be referred to understand the application of federated learning proxy service in the context of O-RAN architecture elements.

The base / shared AI/ML model, which typically begins with randomly initialized parameters or is pre-trained on a non-sensitive public dataset, is prepared at a central server (e.g. O-Cloud or SMO or Non-RT RIC). To enable interactions of the consumers, the FL proxy service exposes a secure endpoint.

1. The FL proxy service consumer (Data owner) provides the raw data for training, FL parameters, communication parameters for the Central FL server (model consumer) and optionally, the communication parameters of itself for a callback to the FL proxy service.
2. Using the Central FL server communication parameters, the FL proxy service requests for the base model.
3. The Central FL server provides the base model using which the FL proxy service performs training and updates to the model weights based on the raw data received from the service consumer to generate a partial AI/ML model. Also, a session is created by the FL proxy service, to distinguish the different consumers supported by the service.
4. The session identifier is sent to service consumer, if consumer communication parameters are available.
5. The FL proxy service sends its local model updates (partial AI/ML model) to the Central FL server.
6. The Central FL server waits for all inputs and aggregates the received model updates to update the shared model and generate an aggregated model. Common aggregation methods include averaging the gradients or weights. The aggregated model is sent to the FL proxy service, if further training is needed based on configured performance / convergence criteria (e.g., reaching a target accuracy, stabilization of loss on a validation set, or completing a set number of communication rounds).
7. The process of local training, sending model updates, and aggregating updates is repeated for multiple iterations until the model converges or achieves satisfactory performance.
8. Once the model meets the configured performance criteria, the training complete indication is sent to the data owner, and the aggregated model can now be deployed in the O-RAN network for real-time applications such as traffic prediction, resource allocation, and anomaly detection.

3.2.3 Impact on AI/ML Models

FL offers significant benefits to AI/ML model in terms of data privacy, security, scalability. However, it also introduces challenges related to interoperability, temporal characteristics, data quality. Some of the key impacts are mentioned below.

- Data owners retain control over their data and do not have to share raw data with external entities thus ensuring privacy.
- Datasets and parameters of local entities must be interoperable with other nodes to ensure seamless integration and model updates.
- Characteristics of datasets may change over time, requiring the model update (re-training) process to account for temporal characteristics and heterogeneity.

3.2.4 Verification and Auditing

To verify the operation of federated learning (FL) in an O-RAN environment, below are some of the verification steps:

- Set up a testbed that simulates the O-RAN components such as, RICs (Non-RT and Near-RT), O-DUs and O-RUs. This includes setting up the base/shared AI/ML model with integrated FL algorithms and hyper-parameters at the central server (Non-RT RIC/ Near-RT RIC) for the intended use cases such as traffic prediction, resource allocation, and anomaly detection in the test bed environment.
- Distribute the shared model to the participating entities which could be the O-DU, O-RU.
- Monitor the local training process to ensure each entity updates the received base/aggregate model using its own data, logging training iterations (epochs) and validating that model parameters are updated correctly
- After local training, the model update objects are sent back to central server contains only gradients or weights.
- Compare the model output trained using FL against a baseline model trained on the raw dataset for an intended use case. This helps to validate the entire federated learning process, from model distribution and local training at O-DUs/O-RUs to the aggregation of updates at the central server, within the simulated O-RAN environment is performing as desired.

3.2.5 Performance and Efficiency

- Training models locally can put extra strain on devices, may potentially slow down device performance. Devices with limited resources may struggle to keep up with the demands of local model training.

- Sending frequent model updates to central server and receiving the aggregated model potentially impacting the responsiveness of real-time applications that rely on FL for decision-making. As more devices join in, the need for computing power and storage grows, which can be challenging if not managed well. Additionally, coordinating these distributed processes can be complex and resource intensive at the central server.

- Coordinating training across multiple entities varying in compute power, bandwidth, availability for training, data characteristics, and even software versions present numerous operational challenges. Addressing these

challenges through standardized protocols is essential for creating a robust and scalable shared machine learning model.

3.3 Differential Privacy

3.3.1 Concept

Differential privacy provides a rigorous and quantifiable guarantee of privacy in data analysis. It achieves this by introducing calibrated noise into the data to preserve the privacy of individual data elements.

Differential privacy achieves this by introducing calibrated noise into the data to preserve the privacy of individual data elements. Unlike simple obfuscation, which may offer vague or unquantifiable privacy, differential privacy provides a mathematical guarantee that the outcome of any analysis will be (almost) the same whether a specific individual data element is included in the dataset.

The formal definition of differential privacy [24] is:

> *A randomized function K gives ϵ-differential privacy if for all data sets D_1 and D_2 different on at most one element, and all S in Range(K),*
>
> $$Pr[K(D_1) \in S] \leq exp(\epsilon) \times Pr\,[K(D_2) \in S]$$

Informally, it states that the output of the differentially private algorithm is almost the same regardless of the presence of a single data point. In practice, such differential privacy can be approximated by output perturbation mechanisms where the output is revealed after adding noise generated with a certain probability distribution that ensures statistical accuracy of the output while preserving data privacy.

To enhance utility of differential privacy algorithms, some relaxations on the definition have been made, e.g., (ϵ,δ) DP, Concentrated DP [25]. Furthermore, in order not to rely on a centralized/trusted curator in running a DP protocol, multi-party versions of DP have been considered [26] [27]. Research and industry are also focusing on achieving strong privacy guarantees using adaptive compositions in differential privacy [4].

3.3.2 Application in O-RAN

Data collection and use in O-RAN has different privacy implications depending on the use cases. Data may be consumed by O-RAN internal network elements (e.g., SMO, Non-RT RIC, Near-RT RIC) for statistical analyses to enhance network energy consumption, resource scheduling, mobility optimization, to name a few. The differentially private data may also be shared with external entities (e.g. Y1 consumers) to support developing advanced algorithms (e.g., AI/ML models) and/or performing analytics.

Differential privacy is an effective and versatile technique to preserve data privacy for various applications in O-RAN while ensuring robust privacy guarantees. In particular, the DP technique is useful for those applications where statistical properties of data

are of importance rather than individual data or data analytics involves confidential information of the operator and its customers. Examples of such data include the number of active users, antenna configuration, physical network topology, performance measurement results. Privacy margin (i.e., ϵ,δ) can be applied differently depending on the nature of data (e.g., personally identifiable information, operational configuration) and applications (e.g., internal or external to MNO).

Establishing data classification and appropriate ϵ,δ values would require a collaborative effort involving standardization bodies and industry working groups. For O-RAN specifically, relevant working groups within the O-RAN ALLIANCE (e.g., those focused on security, Non-RT RIC, or SMO) would likely be involved. It's a critical area for future research and standardization efforts to ensure consistent and effective privacy protection within the O-RAN ecosystem.

3.3.3 Impact on AI/ML Models

Differential privacy is an effective technique to ensure data privacy in AI/ML and can be applied to different stages of AI/ML training and inference. For instance, DP noise can be added to the raw training data, model weights, gradients, or objectives/loss. While DP is efficient compared to other privacy technologies based on cryptographic algorithms in general, it introduces overhead in terms of memory and computation especially when it is used in model updates. Also, DP has different implications on data privacy and AI/ML model utility.

If DP noise is applied to the input data, it would be hard to control the model's privacy guarantees. Whereas, if DP is applied to later stages of AI/ML training, stronger trust assumption is required in distributed or decentralized training environments. In federated learning, if DP is applied by the third-party aggregator/server, individual clients contributing model training need to trust the aggregator/server. Leveraging secure aggregation protocol and/or a secure environment (e.g., Trusted Execution Environment) can relax the trust assumption.

The optimal stage for applying DP is dependent on the specific use case, the desired privacy-utility trade-off, and the trust model. However, based on current understanding and best practices, the recommendation considering the application of DP to model updates or gradients during the aggregation phase in federated learning settings (with joint FL + DP approach). This approach generally offers a better balance between maintaining model utility and providing strong privacy guarantees compared to applying DP directly to raw input data.

3.3.4 Verification and Auditing

Verifying and auditing differential privacy guarantees within O-RAN requires a multi-layered approach that ensures both mathematical correctness and operational compliance. The verification process must address the unique challenges of distributed network architecture while maintaining the rigorous privacy guarantees that differential privacy promises.

3.3.4.1 Privacy Budget Tracking and Management

The privacy loss or privacy budget (ε) represents the fundamental measure of privacy loss in differential privacy implementations[63], effective privacy budget management requires continuous monitoring across all data access points. Establishing the total budget and its component allocation for O-RAN involves a strategic process of balancing regulatory requirements, data sensitivity, desired utility, and the assessed threat model, often via an iterative process of allocation, testing, and refinement. In O-RAN deployments, this involves:

- **Automated Privacy Accounting Systems**: The O-RAN ALLIANCE Security Specifications emphasize the need for comprehensive logging mechanisms. For differential privacy, this translates to maintaining immutable records of:
    - Cumulative privacy expenditure (ε and δ values) across all rApps/xApps
    - Query-specific privacy consumption with timestamps
    - Privacy budget allocation per O-RAN component (SMO, Non-RT RIC, Near-RT RIC)
- **Composition Theorem Implementation**: As established [64], the composition theorem states that total privacy loss after k queries equals $\Sigma\varepsilon_i$. The O-RAN architecture must implement strict enforcement mechanisms to ensure this cumulative loss never exceeds predefined thresholds.

3.3.4.2 Verification Methods and Tools

- **Empirical Verification Frameworks:** IEEE P3156 Standard for Privacy-preserving Computation Platforms [65] establishes requirements for verifying privacy guarantees through statistical testing. For O-RAN implementations, this involves:
    - **Neighboring Dataset Testing:** Creating pairs of datasets differing in exactly one record and verifying that algorithm outputs satisfy the claimed (ε,δ)-differential privacy bounds.
    - **Statistical Hypothesis Testing:** Using χ^2 tests and Kolmogorov-Smirnov tests to verify that output distributions match theoretical expectations within a predefined statistical significance level, accounting for practical deviations from a perfect theoretical fit. [66].
    - **Attack Simulation:** Implementing membership inference [79] and reconstruction [80] attacks to validate privacy guarantees under adversarial conditions [66].

  The 3GPP SA3 [67] emphasizes the importance of empirical validation for privacy-preserving AI/ML implementations.
- **Formal Verification Approaches:** ETSI GS NFV-SEC 031 [68] defines security assurance frameworks that can be extended to differential privacy verification:

- **Automated Theorem Proving**: Using tools like EasyCrypt or CertiPriv to formally verify differential privacy implementations
- **Type System Verification**: Implementing type systems that statically verify privacy budget consumption at compile time
- **Proof-Carrying Code**: Requiring differential privacy implementations to include machine-checkable proofs of their privacy guarantees

3.3.4.3 Auditing Infrastructure

- **Compliance Auditing Framework:** The GSMA's Network Equipment Security Assurance Scheme (NESAS) [69] provides a baseline for security auditing that should be extended for differential privacy:

  - **Third-Party Certification:** Independent auditors certified in differential privacy implementations should verify:
    - Correct noise mechanism implementation (Laplace, Gaussian, or exponential)
    - Proper sensitivity calculation for each query type
    - Privacy budget tracking accuracy
    - Compliance with configured privacy policies
  - **Continuous Monitoring:** ETSI MEC specifications [70] emphasize continuous security monitoring, which for differential privacy includes:
    - Real-time privacy budget consumption dashboards
    - Automated alerts for anomalous privacy loss patterns
    - Periodic re-verification of noise generation mechanisms

- **Standardized Audit Trails:** ITU-T Recommendation Y.3652 [71] on Big Data - Overview and Requirements for Data Preservation specifies audit trail requirements applicable to differential privacy:

  - **Immutable Logging:** All privacy-sensitive operations must generate tamper-proof logs containing:
    - Timestamp and unique transaction ID
    - Privacy parameters used (ε, δ, sensitivity)
    - Data sources accessed
    - Requesting entity identification
    - Output noise characteristics
  - **Audit Log Analysis Tools:** Automated tools to analyze audit logs for:
    - Privacy budget violation attempts
    - Unusual query patterns suggesting privacy attacks
    - Compliance with regulatory requirements (GDPR – General Data Protection Regulation, CCPA - California Consumer Privacy Act)

3.3.4.4 O-RAN Specific Verification Considerations

A key concern is Multi-Vendor Interoperability Testing, which necessitates verifying that differential privacy implementations from diverse vendors maintain consistent privacy guarantees across the ecosystem. This includes ensuring privacy budget accounting remains accurate when data flows across vendor boundaries and rigorously testing privacy preservation in multi-vendor federated learning scenarios.

Furthermore, Real-Time Verification Requirements are paramount, especially for Near-RT RIC applications that operate within stringent latency constraints. To meet these demands, the hardware-accelerated verification techniques, such as FPGA-based privacy budget trackers, GPU-accelerated statistical testing, and specialized ASICs for noise generation verification can be used. These approaches are crucial for enabling continuous and efficient privacy assurance in dynamic O-RAN environments.

3.3.5 Performance and Efficiency

The integration of differential privacy (DP) into O-RAN networks introduces performance impacts that require careful characterization and optimization to balance service quality with privacy protection.

Computational Overhead: DP implementations incur computational overhead primarily from noise generation and sensitivity calculations. Different noise mechanisms have varying costs: the Laplace mechanism requires additional CPU cycles for random number generation and sampling, the Gaussian mechanism often demands slightly more, and the Exponential mechanism can incur a more significant computational cost for complex queries.

Hardware acceleration, such as Intel's RDRAND and RDSEED instructions, can substantially reduce noise generation overhead. Sensitivity analysis, crucial for calibrating noise, can also add overhead, especially when calculated dynamically. For machine learning, DP-SGD can significantly increase training time due to per-example gradient computation and clipping operations. Private aggregation in federated learning also adds overhead, though hardware acceleration can mitigate some of these impacts.

Memory Requirements: DP necessitates additional memory for privacy budget tracking (e.g., for active queries and historical audit trails), noise distribution parameters, and, in AI/ML contexts, for gradient buffers during training. Effective caching strategies for pre-generated noise and query results can also demand significant memory resources.

Latency and Throughput Impact: DP introduces additional latency to query processing, which is noticeable even for simple statistical queries and more significant for complex aggregations. This is a critical consideration for Near-RT RIC applications with stringent latency constraints. Throughput can also be degraded, with the extent depending on the chosen privacy level and implementation specifics.

Privacy-Utility Trade-offs: A fundamental aspect of DP is the trade-off between privacy and data utility. Applying DP can lead to a reduction in the accuracy of network KPIs, predictive models, and an increase in false positive rates for anomaly detection, with the impact varying based on the privacy budget (ε). However, techniques like Rényi DP can offer improved utility and more efficient budget utilization compared to traditional (ε,δ)-DP. Adaptive privacy budget allocation can also enhance utility for equivalent privacy guarantees.

Scalability Analysis: DP's scalability in O-RAN involves considerations for both horizontal and vertical scaling. Privacy budget management scales proportionally with the number of queries, while distributed DP and federated learning can increase communication and bandwidth requirements [34]. Vertical scaling can be optimized through GPU acceleration for batch noise generation or FPGA implementation for real-time tracking.

Comparative Performance Analysis: Compared to other privacy-preserving technologies, DP offers a balance: it generally incurs noticeable latency and throughput overheads and some accuracy loss but provides excellent scalability. This contrasts with homomorphic encryption and secure multi-party computation, which typically have higher overheads but offer minimal accuracy loss, and K-anonymity, which has comparable overheads but often weaker privacy guarantees.

Optimization Strategies: Optimization strategies include implementation optimizations like batch processing to amortize noise generation and adaptive sampling for large datasets, with hardware acceleration offering significant speedups. Architectural optimizations involve edge-based DP to distribute privacy operations, hierarchical privacy application with varying ε values across network layers, and privacy-aware orchestration for dynamic resource allocation.

3.4 Secure Multi-Party Computation

3.4.1 Concept

Secure multi-party computation (SMPC or simply MPC) is a cryptographic technique allowing multiple parties to jointly compute a function (e.g., statistical aggregations, or machine learning model training) over their private inputs. In fact, MPC enables collaborative computation where privacy is preserved by design. More concretely, only the output of the function can be learnt by all or some of the parties, while the intermediate data or input from each party is kept private in the evaluation of the function.

This technology is increasingly used in edge cloud computations and for training machine learning models [5] with the aim to preserve data privacy. For example, in cellular systems, multiple operators can jointly construct a shared interference map without revealing their RF measurement data. As a potential example, MPC in O-RAN allows multiple operators to jointly compute insights such as common coverage gaps, shared traffic congestion patterns, or aggregated quality of experience (QoE) metrics for specific geographical areas without exposing raw network performance data.

MPC has been extensively studied since the introduction of 2PC [28]. The basic idea is to let all participants of the protocol to evaluate a function locally on their private data and then produce output collectively without revealing private data. Secret sharing [29] and Oblivious Transfer [30] [31] are considered fundamental building block in designing a secure MPC protocol. Over the past decades, various protocols [GMW[32], BGW[33], BMR[34]] were developed under semi-honest adversary model and malicious adversary model, while the main bottleneck to their practical usage was the computation and communication complexity. Authentication of computation further increases the overhead.

Efficient MPC protocols have been developed to reduce communication, some of which leverage homomorphic encryption (e.g., SPDZ [35]) to minimize the interactions in evaluating a function. More recently, application of MPC to AI has been studied as a way to enhance the privacy of training data [36] [37] [38] [39].

For practical, multi-party implementations requiring malicious security and arithmetic operations, SPDZ is generally the best choice. It achieves high online-phase efficiency by pre-computing correlated randomness in an offline phase, offering robust security without sacrificing performance, making it superior to GMW (best for 2-party Boolean operations) or BGW (semi-honest arithmetic operations) for complex, real-world applications with malicious adversaries.

3.4.2 Application in O-RAN

The O-RAN architecture is based on openness and interoperability across various components operated by different vendors, e.g. even different MNOs. Although disaggregation is an important feature of O-RAN that enhances flexibility and innovation, it introduces distribution of trust across different organizations. In other words, the architecture must be able to deal with private as well as business-sensitive data across trust boundaries.

While MPC offers strong data privacy, its application may not be easily justified for single MNO scenarios due to the complexity and overhead it would introduce to the system. Meanwhile, when an MNO outsources computing resource from Cloud Service Providers (CSPs), it may want to reduce the risk of single point of failure in performing tasks involving privacy-sensitive data. MPC would be particularly useful in multi-tenant cloud environments. For example, subscriber' location information, mobility pattern and traffic pattern are all privacy sensitive information, and hence any optimization or feature utilizing such information would better leverage MPC to protect subscriber privacy.

In multi-MNO deployment scenarios such shared O-RU, Multi-Operator RAN (MORAN) or Multi-Operator Core Network (MOCN), MNOs may want to make some cooperative decision (e.g., scheduling, resource allocation), without sharing potentially confidential data that would be used as input to the decision process.

More concretely, several promising use cases for MPC in O-RAN include:

- Multi-tenant RAN optimization: In shared deployments such as MORAN, MOCN or shared O-RU, operators need to reveal highly sensitive business data

(e.g., KPI, mobility patterns, traffic load) to optimize spectrum usage, interference mitigation and in general better resource allocation. MPC can enable such collaboration while keeping each operator's raw data private.

- Privacy-preserving federated analytics: If multiple RIC instances, apps within a RIC or different network domains seek to jointly compute network-wide KPIs, MPC can support aggregation of sensitive data without ever transferring or revealing it across different trust domains.
- Outsource computation from untrusted clouds: In scenarios where resource-constrained O-RAN components outsource computation from external cloud providers, MPC can be used to provide security and privacy by using multiple clouds. This approach limits the risk of data exposure even if one of the compute-providers is compromised.
- Secure AI/ML model training across tenants: Multi-operator cooperative model training is challenging if such training requires private and business sensitive data. MPC potentially enables a setting in which multiple MNOs can build quality AI/ML models using rich data sets without revealing their private data.

3.4.3 Impact on AI/ML Models

MPC enables multiple O-RAN components to collaborate on model training or inference without revealing raw sensitive data. In O-RAN, this can potentially be relevant for multi-tenant/shared deployments (e.g., shared O-RU, MORAN/MOCN), cross-operator analytics, and partially trusted clouds.

Model design implications: MPC typically operates over integers using fixed-point arithmetic. This requires choosing a scaling factor and performing quantization. This can affect convergence stability during training (e.g., gradient explosion/vanishing) and the numerical accuracy of inference. Non-linear operations such as ReLU, max, argmax, comparisons are significantly more expensive than linear layers. Therefore, architectures with fewer and simpler non-linearities, average pooling instead of max pooling, folded batch-norm and linear-heavy blocks tend to be preferred. Hybrid approaches that evaluate linear layers with arithmetic sharing and non-linear layers with comparison/Boolean circuits are common in literature [36][37][38][39].

Training vs. inference: MPC-based inferencing is generally more practical than full training because back-propagation multiplies the number of non-linearities and communication rounds. Practical deployments often use MPC for (i) privacy-preserving inferencing, (ii) secure aggregation of model updates in FL, or (iii) training of small/medium models with careful circuit co-design. When training is required, gradient clipping, normalization, and mixed precision fixed point are commonly used.

Privacy/utility trade-offs: MPC prevents data leakage during computation, but it does not by itself protect against information leakage from the published outputs (e.g., trained models). When models or detailed predictions are exported, therefore; other

complementary safeguards (e.g., DP on updates/outputs, output restriction or model access controls) remain necessary. On the other hand, keeping outputs confined to the parties or producing only aggregated KPIs can reduce residual leakage.

Interoperability and data readiness: As with FL, feature schemas, units, sampling intervals, normalization must be aligned across parties. Concept drift, referring to the phenomenon where the statistical properties of the target variable or input features change over time in unforeseen ways (e.g., evolving network traffic patterns, user behavior, or radio propagation characteristics in O-RAN, leading to reduced model reliability), and seasonal effects require periodic re-training or re-sharing of preprocessed statistics. Secure processing (e.g., secret-shared normalization parameters) avoids accidental leakage.

O-RAN-specific benefits: MPC enables operators/apps jointly compute cross-domain KPIs, interference maps, or policy suggestions without ever exposing per-operator counters, UE location traces, or proprietary configuration data. This fits settings with limited mutual trust and contractual separation of duties.

3.4.4 Verification and Auditing

This section details the considerations for MPC vendors to ensure the robust verification and auditing of MPC solutions. To verify MPC in O-RAN, the focus is on correctness, security (per threat model), and operational KPIs in realistic topologies.

Testbed setup. One could emulate parties (e.g., two MNOs and a neutral orchestrator) mapped to Non-RT RIC/SMO, Near-RT RIC and data producing E2 nodes. Afterwards, a workload must be selected. For example, one could choose (i) privacy-preserving inferencing (e.g., traffic classification), (ii) secure aggregation for FL, or (iii) joint KPI computation. Then, we need to pick a numeric format (fixed-point scale, clipping bounds), compile the model/function to an MPC representation (arithmetic/Boolean or hybrid), and document graph size and non-linear ops.

Correctness checks. One must compare MPC outputs to plaintext baselines on held-out data within an agreed tolerance (e.g., absolute/relative error from fixed-point quantization). Moreover, we must unit-test non-linear subroutines (ReLU, compare, argmax, max/average pooling) against edge cases. Lastly, determinism/reproducibility must be verified via fixed seeds and version-locked dependencies.

Security checks (per adversary model). For the semi-honest setting, one must ensure there is no plaintext logging, verify transcripts contain only shares/garbled values, and confirm there is no auxiliary channels (e.g., debug endpoints). In the malicious setting, it is crucial to enable protocol integrity mechanisms (e.g., MAC-authenticated shares as in SPDZ-style protocols), consistency checks on preprocessing materials, and more. One must validate that any detected deviation aborts safely and leaves no partial leakage. It is recommended to document the corruption threshold and recovery behavior (e.g., share refresh, proactive secret sharing) and test drop/rejoin scenarios.

Audit artifacts. It is recommended to preserve and maintain artifacts such as threat model and protocol choice rationale, circuit/model hashes, parameter files, preprocessing integrity proofs, config and software commit IDs and test reports.

3.4.5 Performance and Efficiency

MPC performance is dominated by communication rounds, typically caused by non-linear operations with additional costs from fixed-point arithmetic and preprocessing.

Efficiency challenges:

- Communication-bound execution: non-linear operations such as comparison, max/argmax and conditional branches incur multiple rounds. Therefore, performance is sensitive to RTT between parties.
- Preprocessing overhead: Many MPC protocols, e.g. SPDZ, rely on offline generation of multiplication materials (e.g., Beaver triples). Generating, distributing and storing these materials adds latency, memory and practical complexities.
- Fixed-point costs: Scaling, truncation, and overflow handling add instructions and care in training loops.
- Scalability with parties: scalability has a direct relation with the threat model. For example, dishonest-majority protocols often have higher per-op costs. On the contrary, honest-majority protocols are faster but require stronger deployment assumptions with respect to security.

Optimization to improve efficiency:

- AI Model/circuit co-design: MPC works best with linear-heavy architectures. On could replace max with average pooling, fuse batch-norm into weights, use piecewise-linear or low-cost activations, prune and quantize aggressively, and cap sequence lengths/window sizes for RAN time-series.
- Protocol selection & hybrids: It is recommended to use secret-sharing based arithmetic for linear layers, Boolean/garbled circuits for comparison and leverage OT-extensions.
- Batching and vectorization: One should operate on secret-shared tensors with batched General Matrix Multiplication (GEMM), exploit Single-Instruction Multiple-Data (SIMD) and Basic Linear Algebra Subprograms (BLAS) backends where the field/modulus permits.
- Preprocessing pipelines: it is recommended to amortize offline materials generation, cache and reuse where safe, and parallelize preprocessing across cores/nodes.
- Network placement: it is recommended to co-locate MPC parties (e.g., within the same region) to reduce RTT and use dedicated high-bandwidth links between clouds.
- Operational tactics: it is more efficient to run model inferencing with MPC and secure aggregation in FL when full MPC training is infeasible.

When to choose MPC in O-RAN:

MPC should be chosen where strong inter-party privacy with limited mutual trust joint computation is needed. MPC can potentially be useful when in multi-tenant/shared RAN scenarios in which revealing raw counters/UE traces is unacceptable, but joint optimization brings value. Lastly, MPC can be an alternative to hardware-based protections (TEEs) in situations where TEEs alone are insufficient (e.g., regulator or business constraints), and homomorphic encryption would be too costly for the target function.

Overall, MPC can potentially offer privacy for collaborative analytics in O-RAN. Achieving practical performance requires co-design of models, protocols, and deployment topology in combination with complementary safeguards (e.g. DP) to mitigate residual output leakage.

3.5 Unified data representation for anonymization

3.5.1 Concept

This alternative technique can be used independently from the techniques mentioned in previous sections. As explained below, this mechanism provides privacy preservation, storage space requirement reduction, unified and possible data-agnostic AI/ML algorithms usage and possibilities of bulk predictions.

By converting diverse data types into a uniform matrix format, we can simplify the underlying analytical requirements. This approach leverages the ability to represent various forms of information as a consistent structure, facilitating a more efficient and unified processing pipeline.

A two-dimensional digital image is a matrix of pixels, with each pixel represented as a number, or a small set of numbers, that describe some property of the pixel, such as its brightness (the intensity of the light) or its color. The numbers are arranged in a matrix of rows and columns that correspond to the vertical and horizontal positions of the pixels in the image. If all different kinds of data get represented in the form of images and videos, only image analytics and video analytics would be required to process large amounts of information. Examples from experiments shown in Annex A of the present document give more details about this technique. Figures in Annex A also demonstrate how transforming data into images can help privacy preservation.

This concept of converting any kind of data into images and/or videos can not only reduce the variety of data processing and AI/ML models but also preserve privacy and ensure that any consumer of data does not know any privacy sensitive information. Inferences drawn from image and video analytics would need to be translated into relevant actions according to use case.

Published papers in [19] and [20] discuss converting time-series data into images. Authors of [21] and [22] have shown how video prediction can enhance the prediction accuracies for time-series. PM and KPI data collected in wireless networks are time-series data and these measurements are performed periodically according to the

configurations. This kind of data can be collected periodically from multiple cells for various use-cases presented in [21] which use the PM and KPI data for prediction as well as statistical analysis. Any matrix of numbers can be converted into a digital image, where each number represents the properties of the pixel at that location in the image.

Note that even text-based logs can be converted to images, using persistent clustering, which can provide cluster IDs (numbers) which can then be converted to pixel values. Refer to Annex A of the present document for example implementation.

3.5.2 Application in O-RAN

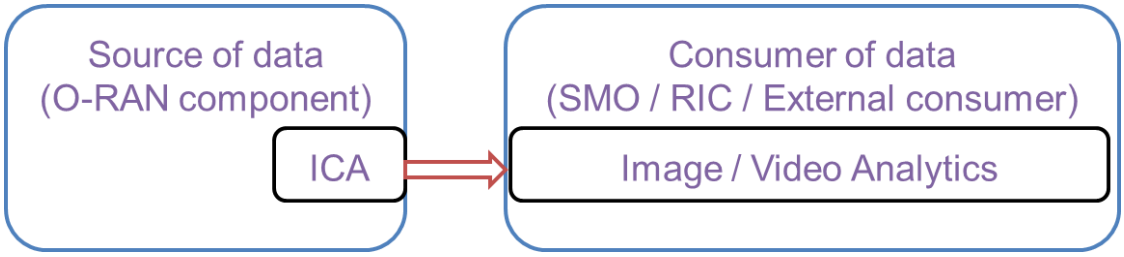


Figure 4: Image Conversion Agent (ICA) to convert data into images at source

Any/all sources of privacy-sensitive data can convert the data into images and/or videos. ICA (Image Conversion Agent) can be implemented in source of data, which can be any O-RAN component, or UEs. This agent can be specific for each source of data and can maintain its own meta-data which can be used to perform the reverse conversion from images to numeric data.

Implementations can be such that this conversion is done selectively only for un-trusted consumers of data. If numeric data is converted into pixel values, as long as the method/formulae used for such a conversion is not known to the consumer, privacy sensitive information cannot be leaked. Note that knowledge of location of privacy sensitive information can also be hidden by using this technique.

This concept is also applicable for CN-RAN architecture, which integrates the functions of the Radio Access Network (RAN) and Core Network (CN). This convergence aims to enhance network efficiency and support advanced applications, particularly as the industry moves towards 6G. In CN-RAN architecture, if all kinds of data including management data and possibly control-plane, user-plane and measurement plane data can be converted to images and videos. Such unified data representation not only helps privacy preservation but can also enable more efficient CN-RAN architecture and enable unified data processing even for metaverse data in 6G.

3.5.3 Impact on AI/ML Models

With this technique, if all data sources convert all data into images and videos, AI/ML models can be more unified and generic in nature. Also, lot of use cases can re-use same models for different purposes. Such generalization and possible re-use of models can reduce a lot of implementation and maintenance efforts while ensuring that even if the models are used from external sources, privacy is always preserved without any loss of data. Data-agnostic AI/ML applications can be developed and the

business logic for interpreting the inferences can be implemented specific to different use cases. Operators can also use image and video analytics provided by 3rd party and even share generated images without worrying about privacy leakage as long as the method and meta-data used to convert data to such images is protected within operator's datacenter.

3.5.4 Verification and Auditing

With this technique, it is sufficient to ensure that the method used to convert data into images and the meta-data is securely stored at the ICA (Image Conversion Agent). As long as the security (secrecy) of this meta-data and the method for conversion is ensured, the consumer of data can by no means figure out any privacy sensitive information. There are various methods for conversion of different kinds of data into images, and periodic images can be considered as frames of a video.

3.5.5 Performance and Efficiency

Following are the advantages of this technique from performance and efficiency point of view:

- Unified data-agnostic implementations of various use-cases and requirements possible for O-RAN.
- Number of AI/ML models to be maintained can be reduced and lot of re-use of models can be made possible by having uniform input data formats in the form of images and videos.

Amount of historic training data which can be fed to AI/ML algorithms for training can be increased using same amount of storage space, while still ensuring no loss of data. Annex A details some experiment results which indicates significant reduction in storage space requirements when raw csv data is converted to images or videos.In 6G, AI/ML based solutions for AR/VR would require image and video analytics. If all other data is also represented in the form of images and videos, it could allow a significant re-use of AI/ML based solutions which are agnostic to the input data and applications can have an abstraction according to the application requirements.

3.6 Towards a Generic O-RAN Privacy Service

A generic privacy service within the O-RAN architecture would offer a unified and adaptable framework to deploy and manage various privacy-preserving techniques (PPTs) discussed in this report. This service could be deployed either as an SMO service (SMOS) or as a capability module within the RIC platform, thereby providing a foundational capability for O-RAN deployments.

This service would act as a central orchestrator, enabling dynamic selection and application of the most suitable PPT or combination thereof based on the specific O-RAN use case, data sensitivity, regulatory requirements, and desired trade-offs between privacy, utility, and performance.

By abstracting the complexities of individual PPTs, it would enable O-RAN network functions and third-party applications (xApps/rApps) to seamlessly integrate privacy protection into their data processing workflows, ensuring that sensitive non-personal data is safeguarded across trust boundaries without requiring deep expertise in each underlying cryptographic or anonymization method. This holistic approach would streamline privacy management, foster compliance, and accelerate the secure deployment of AI/ML-driven innovations in O-RAN. Annex C contains a sample workflow of a generic privacy service that could be used to abstract PPT complexities.

4 Legal and Regulatory Implications

The inherent openness of O-RAN, characterized by its disaggregated architecture, diverse vendor ecosystem, and third-party applications, introduces a complex data privacy landscape that directly impacts the implementation of data privacy-preserving techniques.

O-RAN deployments collect granular user equipment (UE) and network data, encompassing sensitive personal information subject to regulations like GDPR (General Data Protection Regulation), alongside crucial non-personal private data such as network configurations and proprietary algorithms essential for competitive advantage and security. This necessitates the careful selection and application of appropriate privacy-preserving techniques to enable open interfaces and secure data sharing across trust boundaries, mitigating risks to both personal and non-personal data assets.

Understanding the global legal and regulatory landscape is paramount for deploying effective data privacy-preserving techniques in O-RAN. Cross-border data flow restrictions, codified in instruments like GDPR's Chapter V (Regulation (EU) 2016/679) [45] and the Schrems II decision (Data Protection Commissioner v Facebook Ireland and Maximillian Schrems (Case C-311/18) [44]), increasingly extend to strategically important non-personal data.

Compliance demands a comprehensive approach, navigating data protection laws (e.g., GDPR, CCPA), trade secret directives (EU Trade Secrets Directive (2016/943)) [46], national cybersecurity laws (e.g., German IT Security Act (IT-SiG) [47]), and export control regulations (e.g., U.S. Export Administration Regulations (EAR) [48]).

This research report therefore explored the application of privacy-preserving techniques and the development of specific guidelines based on privacy principles [42] for handling all types of sensitive data within the O-RAN ecosystem, leveraging resources from organizations like NIST [41] and ENISA [43] and supporting the development of industry-wide best practices through the O-RAN ALLIANCE to ensure legally compliant and privacy-respecting O-RAN deployments, acknowledging that while these technical solutions are essential enablers, a comprehensive analysis of specific legal and regulatory restrictions would constitute a separate, dedicated study.

4.1 Cross-Border Data Flows

O-RAN's disaggregated architecture, driven by multi-vendor collaboration, third-party analytics, and cloud-based management, inherently leads to cross-border data flows. While these flows are essential for network optimization and innovation, they raise significant legal and privacy concerns.

These concerns are heightened by varying global data protection laws, the potential for foreign government access requests (e.g., under the U.S. CLOUD Act [50]), and challenges in enforcing data subject rights across jurisdictions.

It is important to note that the EU aims to facilitate the free flow of non-personal data within its borders through Regulation (EU) 2018/1807 [49], but this framework does not automatically extend to transfers outside the EU, where other legal considerations, such as trade secret protection [46], may apply.

Privacy-preserving techniques like homomorphic encryption (HE), federated learning (FL), differential privacy (DP), and secure multi-party computation (SMPC) offer potential solutions.

HE reduces the risk of data misuse but doesn't eliminate the need to assess the recipient country's legal framework. FL can comply with data localization requirements, though model updates may still leak information and require the use of differential privacy. DP aims to anonymize data but ensuring sufficient noise to prevent re-identification is crucial. SMPC requires robust data processing agreements with all parties involved.

Regardless of the technique employed, operators and vendors must collaboratively address fundamental legal questions, define clear roles (e.g., data controller and processor), and ensure ongoing compliance with applicable data protection laws (e.g., GDPR [45]), cybersecurity regulations, and trade secret protections across all involved jurisdictions. A risk-based approach, combined with Privacy by Design principles [42], is essential for responsible O-RAN deployments.

4.2 Compliance with Global Data Protection Laws

Deploying O-RAN with privacy-preserving techniques (PPTs) requires a comprehensive strategy that aligns with global data privacy laws, encompassing both personal and non-personal private data.

This strategy necessitates a thorough understanding of applicable legal frameworks, enabling O-RAN stakeholders to navigate the boundaries of acceptable data processing activities and reduce legal uncertainty.

Central to this approach are fundamental principles like Privacy by Design and by Default, Data Minimization, Transparency, Security, and Accountability, operationalized across the O-RAN ecosystem to ensure secure and privacy-respecting data handling for all stakeholders.

To ensure compliant cross-border data transfers, organizations must implement a multi-faceted approach including developing clear privacy policies, conducting

thorough data mapping and risk assessments, and establishing robust contractual agreements based on SCCs [53], BCRs [52] or similar rules.

Furthermore, implementing appropriate security controls, honoring data subject rights (where applicable and extending transparency principles to non-personal data), establishing monitoring and auditing processes, and enforcing strict storage limitation policies are crucial.

This comprehensive strategy, guided by legal frameworks like GDPR [45], DPDPA [54], CCPA [55], CPRA [81], PIPL [82], cybersecurity regulations [51], and trade secret protections [46], ensures the responsible and legally compliant deployment of O-RAN with privacy-preserving techniques.

5 Conclusion

5.1.1 Summary

This research report, produced by the O-RAN Next Generation Research Group (nGRG), delves into privacy-preserving techniques and their application within O-RAN network functions. The core objective is to analyze the impact of these techniques on AI/ML models during training and inference, evaluate their effect on network function performance, and address the crucial aspects of verification and auditing for regulatory and legal compliance. The report specifically focuses on non-personal data within O-RAN, considering use cases such as data shared with third-party applications (xApps and rApps) via the RIC platform (E2 and Y1 interfaces) and enrichment data shared over the A1 interface.

The O-RAN architecture, while fostering openness and innovation, introduces significant data privacy challenges due to its disaggregated nature and the involvement of multiple stakeholders. This creates an increased attack surface, risks of data interception, and third-party vulnerabilities. The report highlights that O-RAN handles vast amounts of sensitive non-personal data, including network metrics, location patterns, and device-specific performance data, which require robust privacy protection. The integration of AI/ML algorithms further complicates matters, with potential threats like data leakages from improper isolation and malicious applications. The report emphasizes the need for solutions that go beyond basic security controls, especially when data crosses trust boundaries to third-party applications.

The report explores five key privacy-preserving techniques: homomorphic encryption (HE), federated learning (FL), differential privacy (DP), secure multi-party computation (SMPC), and unified data representation for anonymization.

For each technique, the report details its concept, application in O-RAN, impact on AI/ML models, verification and auditing methods, and performance and efficiency considerations. For instance, HE enables computations on encrypted data without decryption, crucial for privacy-preserving AI/ML tasks in multi-vendor O-RAN environments. FL allows collaborative model training without raw data exchange, ideal for O-RAN components to jointly develop models while keeping sensitive data

localized. DP introduces calibrated noise to preserve individual data elements' privacy, particularly useful for statistical analyses where aggregate properties are more important than individual data points. SMPC allows multiple parties to jointly compute a function over their private inputs, addressing trust issues in multi-tenant or cross-operator scenarios. Finally, unified data representation for anonymization, such as converting data into images, aims to reduce data processing variety and inherently preserve privacy by obscuring sensitive information from consumers.

Various sections of this report explain in depth the privacy-preserving techniques and their potential applicability to O-RAN, underscoring the careful consideration required for combining these techniques. It also highlights the critical need to weigh performance trade-offs against the target asset and protocol bandwidth. The standardization of these techniques will be keenly evaluated once they reach sufficient maturity.

The legal and regulatory landscape for data privacy in O-RAN is also a critical focus. The report discusses the complexities of cross-border data flows, especially with stringent regulations like GDPR, and the need for compliance with global data protection laws. It underscores that while privacy-preserving techniques offer potential solutions, they do not eliminate the need for careful legal assessment, clear role definitions, and robust contractual agreements. The report advocates for a risk-based approach and the adoption of Privacy by Design principles to ensure legally compliant and privacy-respecting O-RAN deployments.

5.1.2 Future Work

The future work aims to explore the application and optimization of the privacy-preserving techniques addressed in this research report within specific Open RAN use cases, with a strong focus on enhancing AI/ML security and data privacy. This will involve conducting empirical studies to evaluate the effectiveness, performance overheads, and scalability of homomorphic encryption, federated learning, differential privacy, secure multi-party computation, and unified data representation for anonymization in diverse O-RAN environments. Specifically, we will prioritize near-term O-RAN deployments and identify critical privacy vulnerabilities in their AI/ML pipelines and xApp / rApp supply chains.

Furthermore, research will concentrate on developing hybrid approaches that combine multiple techniques to achieve stronger privacy guarantees, while also prioritizing the creation of standardized metrics and methodologies for verifying and auditing the privacy assurances of these technologies within O-RAN, ultimately contributing to robust policy recommendations and industry best practices. A key focus will be on defining concrete architectural guidelines and deployment strategies for these privacy-preserving techniques to address the risks. Additionally, this work will explore the foundational requirements and early integration strategies for privacy-by-design principles relevant to future 6G networks, considering their anticipated data-intensive and AI-driven nature.

References

[1] B. Mao, J. Liu, Y. Wu and N. Kato, "Security and Privacy on 6G Network Edge: A Survey," in IEEE Communications Surveys & Tutorials, vol. 25, no. 2, pp. 1095-1127, Second quarter 2023, doi: 10.1109/COMST.2023.3244674.

[2] Junxue Zhang, Xiaodian Cheng, Liu Yang, Jinbin Hu, Ximeng Liu, and Kai Chen. 2024. SoK: Fully Homomorphic Encryption Accelerators. ACM Comput. Surv. 56, 12, Article 316 (December 2024), 32 pages. https://doi.org/10.1145/3676955

[3] Q. Duan, J. Huang, S. Hu, R. Deng, Z. Lu and S. Yu, "Combining Federated Learning and Edge Computing Toward Ubiquitous Intelligence in 6G Network: Challenges, Recent Advances, and Future Directions," in IEEE Communications Surveys & Tutorials, vol. 25, no. 4, pp. 2892-2950, Fourthquarter 2023, doi: 10.1109/COMST.2023.3316615.

[4] Justin Whitehouse, Aaditya Ramdas, Ryan Rogers, Steven Wu, "Fully-Adaptive Composition in Differential Privacy", in Proceedings of the 40th International Conference on Machine Learning, PMLR 202:36990-37007, 2023.

[5] I. Zhou, F. Tofigh, M. Piccardi, M. Abolhasan, D. Franklin and J. Lipman, "Secure Multi-Party Computation for Machine Learning: A Survey," in IEEE Access, vol. 12, pp. 53881-53899, 2024, doi: 10.1109/ACCESS.2024.3388992.

[6] O-RAN ALLIANCE Technical Specification: "O-RAN Y1 interface: General Aspects and Principles"

[7] IETF RFC 9068: "JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens" https://datatracker.ietf.org/doc/html/rfc9068#name-claims-registration

[8] O-RAN ALLIANCE Technical Specification: "O-RAN Architecture Description"

[9] O-RAN ALLIANCE Technical Specification: "Non-RT RIC & A1/R1 interface: Use Cases and Requirements"

[10] O-RAN ALLIANCE Technical Specification: "Near-RT RIC Architecture"

[11] Brik, B., Chergui, H., Zanzi, L., Devoti, F., Ksentini, A., Siddiqui, M.S., Costa-Pérez, X. and Verikoukis, C., 2023. A survey on explainable ai for 6g o-ran: Architecture, use cases, challenges and research directions. arXiv preprint arXiv:2307.00319.

[12] Gajjar, P., Chiejina, A. and Shah, V.K., 2024, May. Preserving data privacy for ML-driven applications in open radio access networks. In 2024 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN) (pp. 339-346). IEEE.

[13] Pulido-Gaytan, B., Tchernykh, A., Cortés-Mendoza, J.M., Babenko, M., Radchenko, G., Avetisyan, A. and Drozdov, A.Y., 2021. Privacy-preserving neural networks with homomorphic encryption: Challenges and opportunities. Peer-to-Peer Networking and Applications, 14(3), pp.1666-1691.

[14] "Application-Aware Approximate Homomorphic Encryption: Configuring FHE for Practical Use", https://eprint.iacr.org/2024/203.pdf

[15] "Fast Fully Homomorphic Encryption Library over the Torus", https://github.com/tfhe/tfhe

[16] “A FULLY HOMOMORPHIC ENCRYPTION SCHEME”, https://crypto.stanford.edu/craig/craig-thesis.pdf
[17] O-RAN ALLIANCE Technical Specification: “Use Cases Detailed Specification”
[18] O-RAN ALLIANCE Technical Report: “Use Cases Analysis Report”
[19] Wang, Zhiguang, and Tim Oates. “Imaging time-series to improve classification and imputation.” Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015.
[20] Moghaddam, Arya Hadizadeh, and Saeedeh Momtazi. ”Image processing meets time series analysis: Predicting Forex profitable technical pattern positions.” Applied Soft Computing 108 (2021): 107460.
[21] Cohen, N., Sood, S., Zeng, Z., Balch, T.,& Veloso, M. (2020). Visual Time Series Forecasting: An Image-driven Approach. arXiv preprint arXiv:2011.09052.
[22] Zeng, Zhen, Tucker Balch, and Manuela Veloso. ”Deep video prediction for time series forecasting.” Proceedings of the Second ACM International Conference on AI in Finance. 2021.
[23] Python drain3 : https://pypi.org/project/drain3/0.7.5/ : A log template miner
[24] Dwork, C.: Differential Privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12.
[25] Dwork, C., Rothblum, C.: Concentrated differential privacy. CoRR, abs/1603.01887 2016
[26] Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: simultaneously solving how and what. In Advances in cryptology, CRYPTO 2008,451- 468.
[27] Vadhan, Salil. “The Complexity of Differential Privacy”. In Tutorials on the Foundations of Cryptography, 347-450, 2017
[28] A. C. Yao. Protocols for secure computations. 23rd FOCS, pages 160--164. IEEE Computer Society Press, 1982.
[29] Shamir, A. 1979. “How to share a secret”. Communications of the ACM. 22(11): 612–613.
[30] S. Even, O. Goldreich, and A. Lempel, "A Randomized Protocol for Signing Contracts", Communications of the ACM, Volume 28, Issue 6, pg. 637–647, 1985.
[31] Kilian, J. 1988. “Founding Cryptography on Oblivious Transfer”. In: 20th Annual ACM Symposium on Theory of Computing. ACM Press. 20–31.
[32] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. 19th ACM STOC, pages 218--229. ACM Press, 1987
[33] Ben-Or, M., S. Goldwasser, and A. Wigderson. 1988. “Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation (Extended Abstract)”. In: 20th Annual ACM Symposium on Theory of Computing. ACM Press. 1–10
[34] D. Beaver, S. Micali, and P. Rogaway. The round complexity of secure protocols. Proceedings of the twenty-second annual ACM symposium on Theory of computing, pages 503--513. ACM, 1990.

[35] Damgård, I., V. Pastro, N. P. Smart, and S. Zakarias. 2012b. “Multiparty Computation from Somewhat Homomorphic Encryption”. In: Advances in Cryptology – CRYPTO 2012. 643–662.

[36] Sameer Wagh, Divya Gupta, and Nishanth Chandran, "SecureNN: 3-Party Secure Computation for Neural Network Training", PoPETs 2019.

[37] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan, "GAZELLE: A Low Latency Framework for Secure Neural Network Inference", 27th USENIX Security Symposium, USENIX Security 2018.

[38] Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, Laurens van der Maaten, "CrypTen: Secure Multi-Party Computation Meets Machine Learning", https://arxiv.org/abs/2109.00984

[39] Nishant Kumar, Mayank Rathee, Nishanth Chandran, Divya Gupta, Aseem Rastogi, Rahul Sharma, "CrypTFlow: Secure TensorFlow Inference", https://arxiv.org/abs/1909.07814

[40] Sandeepa, C., Siniarski, B., Kourtellis, N., Wang, S. and Liyanage, M., 2024. A survey on privacy of personal and non-personal data in B5G/6G networks. *ACM Computing Surveys*, *56*(10), pp.1-37.

[41] NIST SP 800-188: ”De-Identifying Government Datasets: Techniques and Governance” - https://csrc.nist.gov/pubs/sp/800/188/final

[42] Cavoukian, A., 2009. Privacy by design: The 7 foundational principles. Information and privacy commissioner of Ontario, Canada, 5(2009), p.12.

[43] ENISA – “Guidance and gaps analysis for European standardisation - Privacy standards in the information security context”, https://www.enisa.europa.eu/publications/guidance-and-gaps-analysis-for-european-standardisation

[44] The CJEU judgment in the Schrems II case - https://www.europarl.europa.eu/RegData/etudes/ATAG/2020/652073/EPRS_ATA(2020)652073_EN.pdf

[45] Official Journal of the European Union - On the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) - https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679

[46] Official Journal of the European Union - On the protection of undisclosed know-how and business information (trade secrets) against their unlawful acquisition, use and disclosure - https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016L0943

[47] Second act on increasing the security of IT systems (German IT Security Act 2.0) - https://www.bsi.bund.de/EN/Das-BSI/Auftrag/Gesetze-und-Verordungen/IT-SiG/2-0/it_sig-2-0_node.html

[48] U.S. Export Regulations - https://www.trade.gov/us-export-regulations-0

[49] Official Journal of the European Union - On a framework for the free flow of non-personal data in the European Union - https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32018R1807

[50] U.S. CLOUD (Clarifying Lawful Overseas Use of Data Act) Act - https://www.justice.gov/criminal/media/999391/dl?inline
[51] ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection — Information security management systems — Requirements - https://www.iso.org/standard/27001
[52] EU Commission - Binding Corporate Rules - https://commission.europa.eu/law/law-topic/data-protection/international-dimension-data-protection/binding-corporate-rules-bcr_en
[53] EU Commission – Standard contractual clauses https://commission.europa.eu/law/law-topic/data-protection/international-dimension-data-protection/standard-contractual-clauses-scc_en
[54] The digital personal data protection act (INDIA) - DPDPA - https://www.meity.gov.in/static/uploads/2024/06/2bf1f0e9f04e6fb4f8fef35e82c42aa5.pdf
[55] California Consumer Privacy Act (CCPA) - https://oag.ca.gov/privacy/ccpa
[56] IEEE digital privacy: “Types of Homomorphic Encryption” https://digitalprivacy.ieee.org/publications/topics/types-of-homomorphic-encryption/
[57] NIST. (2016).” Report on Post-Quantum Cryptography. NISTIR 8105”, https://nvlpubs.nist.gov/nistpubs/ir/2016/nist.ir.8105.pdf
[58] Naehrig, M “Can homomorphic encryption be practical? ACM Cloud Computing Security Workshop” https://eprint.iacr.org/2011/405.pdf
[59] Chillotti, Ilaria, Nicolas Gama, Mariya Georgieva, and Malika Izabachene. "Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds." In international conference on the theory and application of cryptology and information security, pp. 3-33. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.
[60] Cheon, Jung Hee, Andrey Kim, Miran Kim, and Yongsoo Song. "Homomorphic encryption for arithmetic of approximate numbers." In International conference on the theory and application of cryptology and information security, pp. 409-437. Cham: Springer International Publishing, 2017.
[61] IACR “ Private Computation on Encrypted Genomic Data” https://eprint.iacr.org/2015/133.pdf
[62] Brutzkus, Alon, Ran Gilad-Bachrach, and Oren Elisha. "Low latency privacy preserving inference." In International Conference on Machine Learning, pp. 812-821. PMLR, 2019.
[63] NIST “Guidelines for Evaluating Differential Privacy Guarantees”, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-226.pdf
[64] Dwork, C., et al. "The Algorithmic Foundations of Differential Privacy", https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf
[65] IEEE Draft Standard for Requirements of Privacy-preserving Computation Integrated Platforms," in IEEE P3156/D2, July 2023 , vol., no., pp.1-28, 1 Aug. 2023. https://dl.acm.org/doi/pdf/10.1145/3428233

[66] Jagielski, M., Ullman, J. and Oprea, A., 2020. Auditing differentially private machine learning: How private is private sgd?. Advances in Neural Information Processing Systems, 33, pp.22205-22216.

[67] 3GPP TR 33.849, "Study on Security Aspects of Artificial Intelligence/Machine Learning for NG-RAN", https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=4080

[68] ETSI "Network Functions Virtualisation; Security Assurance Specification" https://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/031/05.02.01_60/gs_NFV-SEC031v050201p.pdf

[69] GSMA "Network Equipment Security Assurance Scheme (NESAS) Overview", https://www.gsma.com/solutions-and-impact/technologies/security/wp-content/uploads/2023/08/FS.13-v2.3-NESAS-Overview-PDF-2.pdf

[70] ETSI " Multi-access Edge Computing; Phase 2: Use Cases and Requirements"

[71] ITU-T "Big Data - Overview and Requirements for Data Preservation"

[72] O-RAN "O-RAN Security Threat Modeling and Remediation Analysis"

[73] 3GPP TS 23.501, "System architecture for the 5G System (5GS)", https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144

[74] O-RAN ALLIANCE Technical Report: “O-RAN Security Threat Modeling and Risk Assessment”

[75] NIST Internal Report, "Transition to Post-Quantum Cryptography Standards", https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8547.ipd.pdf

[76] Paillier, P., 1999, April. Public-key cryptosystems based on composite degree residuosity classes. In International conference on the theory and applications of cryptographic techniques (pp. 223-238). Berlin, Heidelberg: Springer Berlin Heidelberg.

[77] Niu, C., Wei, B., Huang, Z., Yang, Z., Hong, C., Wang, M. and Wei, T., 2025. SoK: FHE-friendly symmetric ciphers and transciphering. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2025(3), pp.583-613.

[78] O-RAN next Generation Research Group (nGRG) Contributed Research Report, “Scalable and User-Centric RAN Architecture: Service Requirements and Design Considerations”, https://mediastorage.o-ran.org/ngrg-rr/nGRG_RS02_RR_Scalable_and_User-Centric_RAN_Architecture_and_Service_Requirements_v16.pdf

[79] NIST Glossary, "Membership inference attack", https://csrc.nist.gov/glossary/term/membership_inference_attack

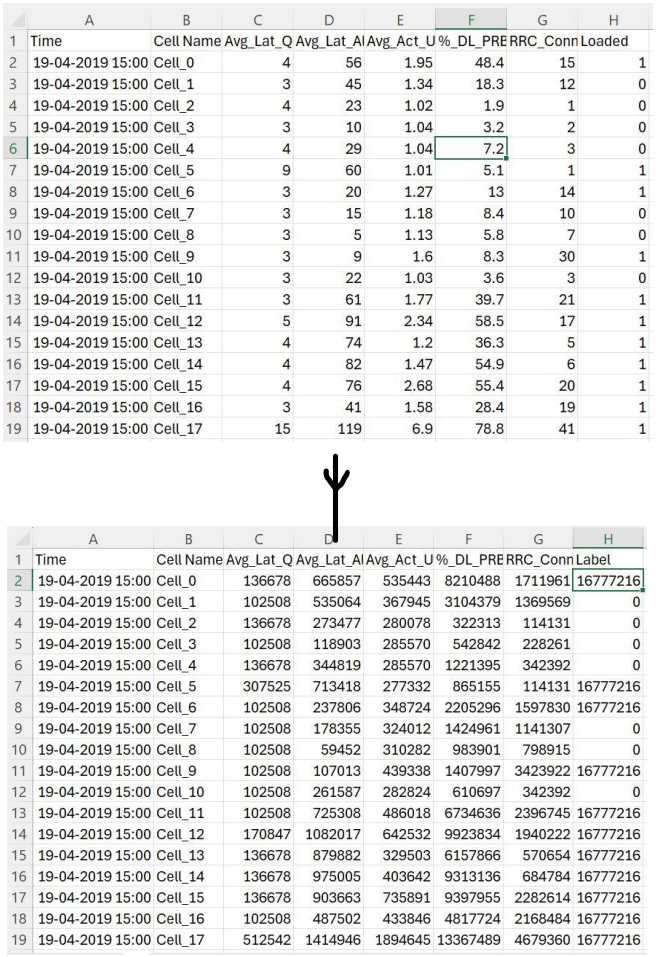
[80] NIST Glossary, "Reconstruction attack", https://csrc.nist.gov/glossary/term/reconstruction_attack

[81] The California Privacy Rights Act of 2020, https://thecpra.org

[82] “Translation: Personal Information Protection Law of the People’s Republic of China” - https://digichina.stanford.edu/work/translation-personal-information-protection-law-of-the-peoples-republic-of-china-effective-nov-1-2021/

Annex A

Examples demonstrating Unified data representation for anonymization



	A	B	C	D	E	F	G	H
1	Time	Cell Name	Avg_Lat_Q	Avg_Lat_Al	Avg_Act_U	%_DL_PRE	RRC_Conn	Loaded
2	19-04-2019 15:00	Cell_0	4	56	1.95	48.4	15	1
3	19-04-2019 15:00	Cell_1	3	45	1.34	18.3	12	0
4	19-04-2019 15:00	Cell_2	4	23	1.02	1.9	1	0
5	19-04-2019 15:00	Cell_3	3	10	1.04	3.2	2	0
6	19-04-2019 15:00	Cell_4	4	29	1.04	7.2	3	0
7	19-04-2019 15:00	Cell_5	9	60	1.01	5.1	1	1
8	19-04-2019 15:00	Cell_6	3	20	1.27	13	14	1
9	19-04-2019 15:00	Cell_7	3	15	1.18	8.4	10	0
10	19-04-2019 15:00	Cell_8	3	5	1.13	5.8	7	0
11	19-04-2019 15:00	Cell_9	3	9	1.6	8.3	30	1
12	19-04-2019 15:00	Cell_10	3	22	1.03	3.6	3	0
13	19-04-2019 15:00	Cell_11	3	61	1.77	39.7	21	1
14	19-04-2019 15:00	Cell_12	5	91	2.34	58.5	17	1
15	19-04-2019 15:00	Cell_13	4	74	1.2	36.3	5	1
16	19-04-2019 15:00	Cell_14	4	82	1.47	54.9	6	1
17	19-04-2019 15:00	Cell_15	4	76	2.68	55.4	20	1
18	19-04-2019 15:00	Cell_16	3	41	1.58	28.4	19	1
19	19-04-2019 15:00	Cell_17	15	119	6.9	78.8	41	1

	A	B	C	D	E	F	G	H
1	Time	Cell Name	Avg_Lat_Q	Avg_Lat_Al	Avg_Act_U	%_DL_PRE	RRC_Conn	Label
2	19-04-2019 15:00	Cell_0	136678	665857	535443	8210488	1711961	16777216
3	19-04-2019 15:00	Cell_1	102508	535064	367945	3104379	1369569	0
4	19-04-2019 15:00	Cell_2	136678	273477	280078	322313	114131	0
5	19-04-2019 15:00	Cell_3	102508	118903	285570	542842	228261	0
6	19-04-2019 15:00	Cell_4	136678	344819	285570	1221395	342392	0
7	19-04-2019 15:00	Cell_5	307525	713418	277332	865155	114131	16777216
8	19-04-2019 15:00	Cell_6	102508	237806	348724	2205296	1597830	16777216
9	19-04-2019 15:00	Cell_7	102508	178355	324012	1424961	1141307	0
10	19-04-2019 15:00	Cell_8	102508	59452	310282	983901	798915	0
11	19-04-2019 15:00	Cell_9	102508	107013	439338	1407997	3423922	16777216
12	19-04-2019 15:00	Cell_10	102508	261587	282824	610697	342392	0
13	19-04-2019 15:00	Cell_11	102508	725308	486018	6734636	2396745	16777216
14	19-04-2019 15:00	Cell_12	170847	1082017	642532	9923834	1940222	16777216
15	19-04-2019 15:00	Cell_13	136678	879882	329503	6157866	570654	16777216
16	19-04-2019 15:00	Cell_14	136678	975005	403642	9313136	684784	16777216
17	19-04-2019 15:00	Cell_15	136678	903663	735891	9397955	2282614	16777216
18	19-04-2019 15:00	Cell_16	102508	487502	433846	4817724	2168484	16777216
19	19-04-2019 15:00	Cell_17	512542	1414946	1894645	13367489	4679360	16777216

Figure 5: Convert raw numeric data to 24-bit pixel values

For experiments, we used some data collected for 1000 cells. After clean-up, 991 cells had good data. The collection was made for latency and load related KPIs for these cells, and using some formula, a label (loaded) was derived. The collection granularity was 15 minutes. **Error! Reference source not found.** illustrates the conversion of numeric data to 24-bit pixel values using min-max scaling. Note that the last column of this data is a label which is either 1 or 0. Corresponding value for 1 in 24 bit representation is 16777216. Subsequently, the columns "Cell Name" and "Time" are also converted into numbers representing pixel values ranging from 0 to (2^{24}-1). We had enough data to split into 677 different files, each file representing one timestamp.

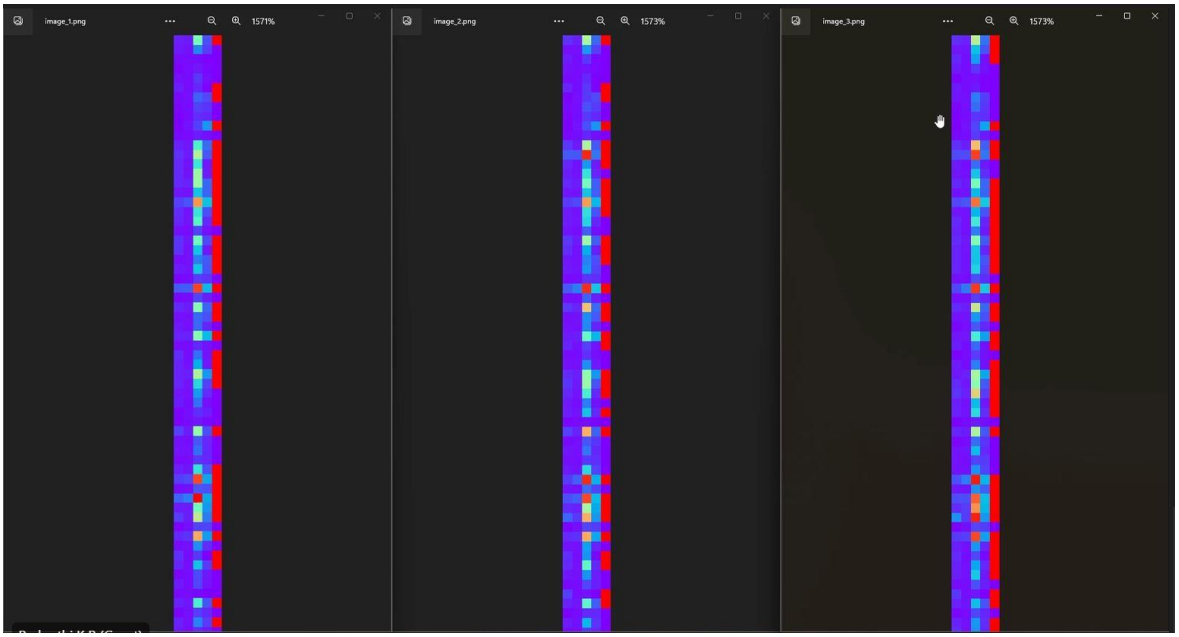


Figure 6: First three parts converted to images

After converting all 677 files into images, we can use each image sequentially as frames of a video. Snapshots of the first 3 images can be seen in **Error! Reference source not found.**. Unless someone knows what data was used to generate such images, and how it was created, no information can be revealed.

Using these images as frames of a video, predicting one frame can give predictions for 5 KPIs and 1 label for 991 cells in one-shot. Moreover, we observe that the original csv file was 31.36MB, whereas the video (avi file) occupied only 130kB.

For text-based data (like syslog), **Error! Reference source not found.** shows the flow-chart with detailed steps. In **Error! Reference source not found.**, “clustering with memory” can be implemented for each column using, for example, python drain3 using persistence enabled. Enabling persistence ensures that when similar log is fed to the same python module, similar logs get same cluster IDs which were assigned earlier.

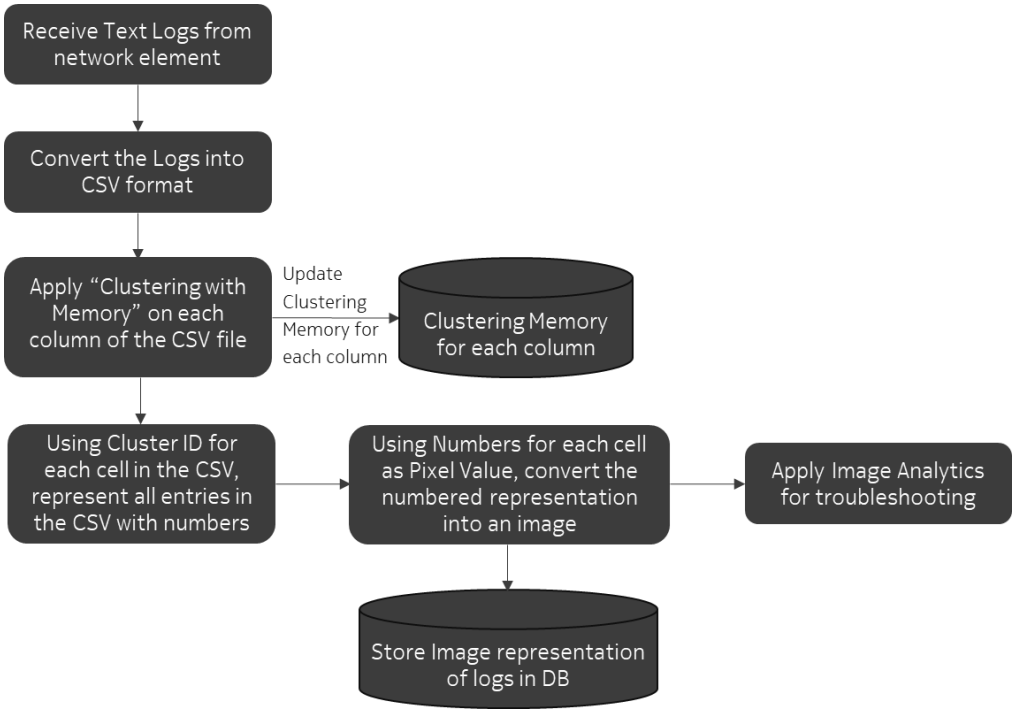
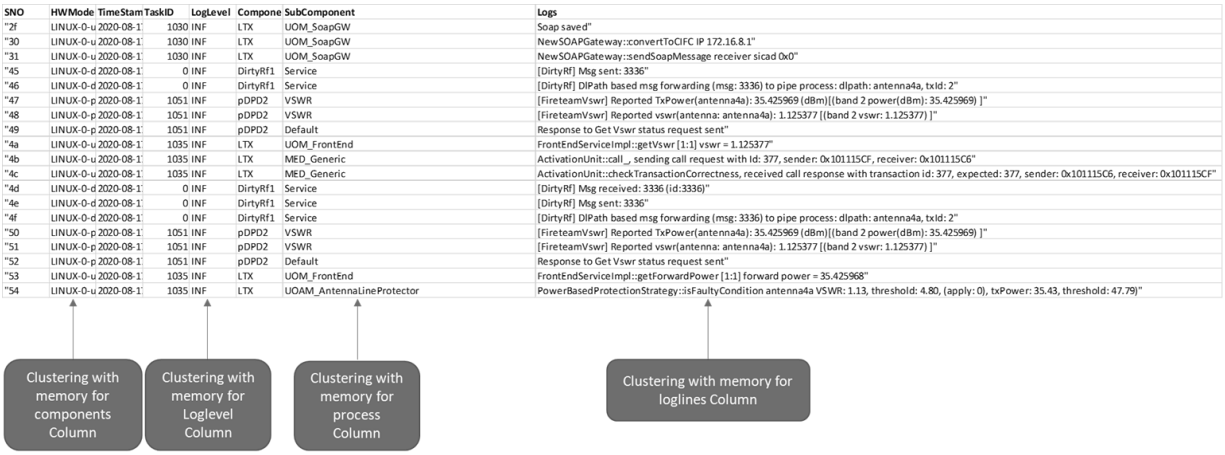


Figure 7: Flow-chart showing steps to convert text data to images



SNO	HWMode	TimeStam	TaskID	LogLevel	Compone	SubComponent	Logs
"2f	LINUX-0-u	2020-08-1	1030	INF	LTX	UOM_SoapGW	Soap saved"
"30	LINUX-0-u	2020-08-1	1030	INF	LTX	UOM_SoapGW	NewSOAPGateway::convertToCIFC IP 172.16.8.1"
"31	LINUX-0-u	2020-08-1	1030	INF	LTX	UOM_SoapGW	NewSOAPGateway::sendSoapMessage receiver sicad 0x0"
"45	LINUX-0-d	2020-08-1	0	INF	DirtyRf1	Service	[DirtyRf] Msg sent: 3336"
"46	LINUX-0-d	2020-08-1	0	INF	DirtyRf1	Service	[DirtyRf] DlPath based msg forwarding (msg: 3336) to pipe process: dlpath: antenna4a, txId: 2"
"47	LINUX-0-p	2020-08-1	1051	INF	pDPD2	VSWR	[FireteamVswr] Reported TxPower(antenna4a): 35.425969 (dBm)[(band 2 power(dBm): 35.425969)]"
"48	LINUX-0-p	2020-08-1	1051	INF	pDPD2	VSWR	[FireteamVswr] Reported vswr(antenna: antenna4a): 1.125377 [(band 2 vswr: 1.125377)]"
"49	LINUX-0-p	2020-08-1	1051	INF	pDPD2	Default	Response to Get Vswr status request sent"
"4a	LINUX-0-u	2020-08-1	1035	INF	LTX	UOM_FrontEnd	FrontEndServiceImpl::getVswr [1:1] vswr = 1.125377"
"4b	LINUX-0-u	2020-08-1	1035	INF	LTX	MED_Generic	ActivationUnit::call _, sending call request with Id: 377, sender: 0x101115CF, receiver: 0x101115C6"
"4c	LINUX-0-u	2020-08-1	1035	INF	LTX	MED_Generic	ActivationUnit::checkTransactionCorrectness, received call response with transaction id: 377, expected: 377, sender: 0x101115C6, receiver: 0x101115CF"
"4d	LINUX-0-d	2020-08-1	0	INF	DirtyRf1	Service	[DirtyRf] Msg received: 3336 (id:3336)"
"4e	LINUX-0-d	2020-08-1	0	INF	DirtyRf1	Service	[DirtyRf] Msg sent: 3336"
"4f	LINUX-0-d	2020-08-1	0	INF	DirtyRf1	Service	[DirtyRf] DlPath based msg forwarding (msg: 3336) to pipe process: dlpath: antenna4a, txId: 2"
"50	LINUX-0-p	2020-08-1	1051	INF	pDPD2	VSWR	[FireteamVswr] Reported TxPower(antenna4a): 35.425969 (dBm)[(band 2 power(dBm): 35.425969)]"
"51	LINUX-0-p	2020-08-1	1051	INF	pDPD2	VSWR	[FireteamVswr] Reported vswr(antenna: antenna4a): 1.125377 [(band 2 vswr: 1.125377)]"
"52	LINUX-0-p	2020-08-1	1051	INF	pDPD2	Default	Response to Get Vswr status request sent"
"53	LINUX-0-u	2020-08-1	1035	INF	LTX	UOM_FrontEnd	FrontEndServiceImpl::getForwardPower [1:1] forward power = 35.425968"
"54	LINUX-0-u	2020-08-1	1035	INF	LTX	UOAM_AntennaLineProtector	PowerBasedProtectionStrategy::isFaultyCondition antenna4a VSWR: 1.13, threshold: 4.80, (apply: 0), txPower: 35.43, threshold: 47.79)"

Figure 8: Example text logs (Clustering with memory)

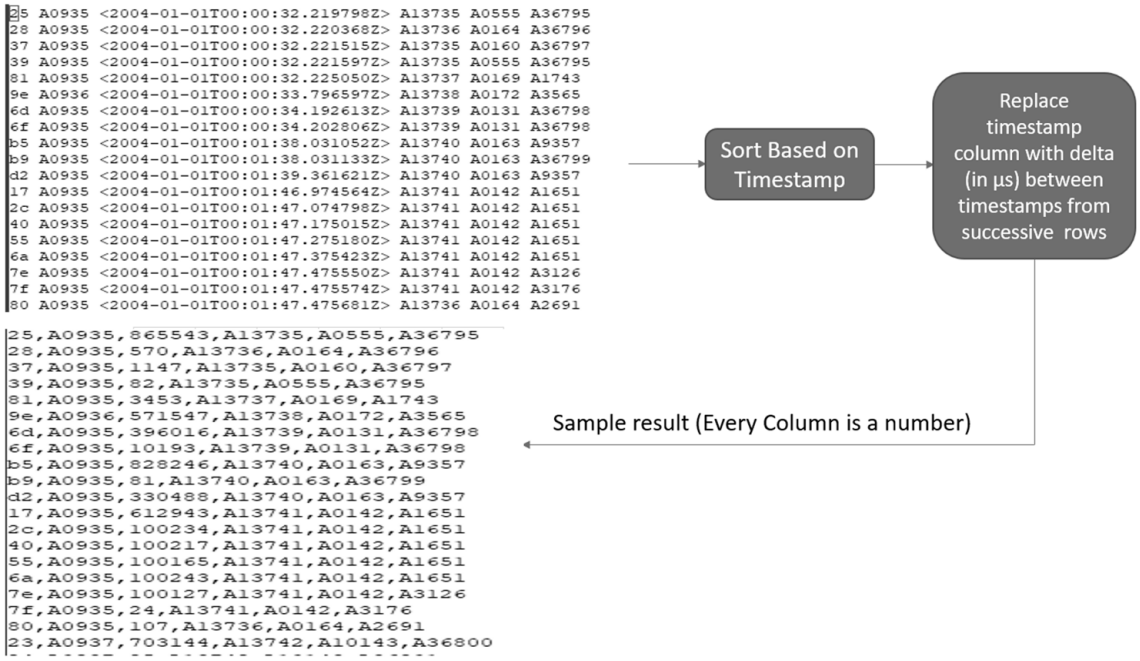


Figure 9: Special handling for the timestamp column

After special handling of timestamp column, as shown in **Error! Reference source not found.**, all columns would have numeric values. The prefix “A” seen here can be removed before applying scaling to make the matrix represent an image with, say, 24 bits per pixel.

Considering an example of text-based troubleshooting logs for a given problem report, comparing these logs to determine whether a new problem report is similar to an old problem report is a challenging task even using AI/ML and NLP techniques. Converting such text-based data into images allows storage of a much larger amount of historic troubleshooting data for more accurate AI/ML based solutions. Our experiments suggest that without any loss of data, using 24 bits per pixel, only one-fifth of the storage is required for storing images representing text-based logs.

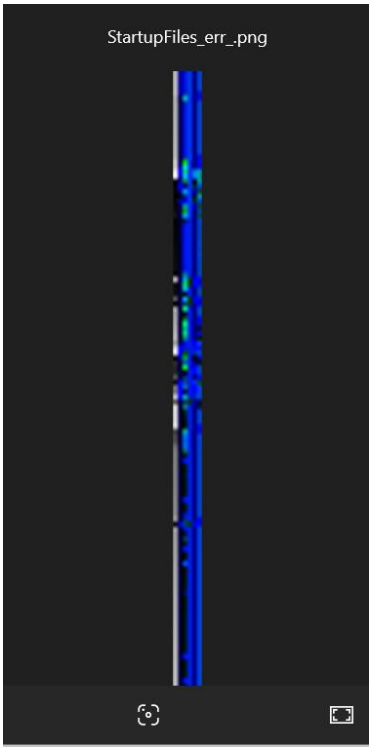


Figure 10: Image representing a text log

Error! Reference source not found. shows an example of an image which represents a text log. This completely conceals privacy, while enabling use of AI/ML models which can work on images. Unlike real-world images, such images cannot be interpreted by humans.

Annex B

Example of FL proxy service usage with E2 nodes, Near-RT RIC and SMO

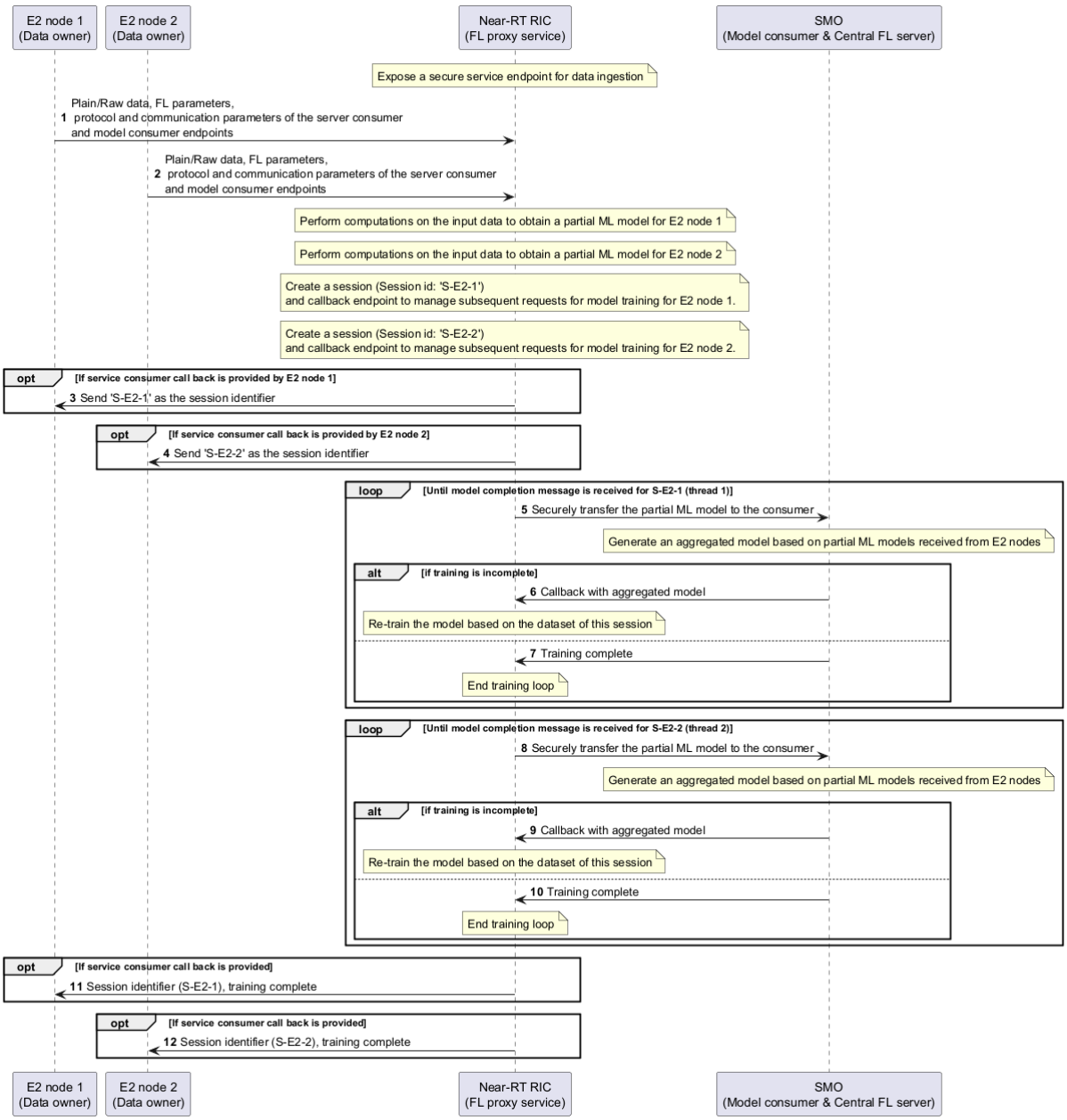


Figure 11: Federated learning proxy service at Near-RT RIC is used by O-RAN E2 nodes, with SMO acting as the Central FL server

Annex C

Generic service abstracting privacy-preserving techniques

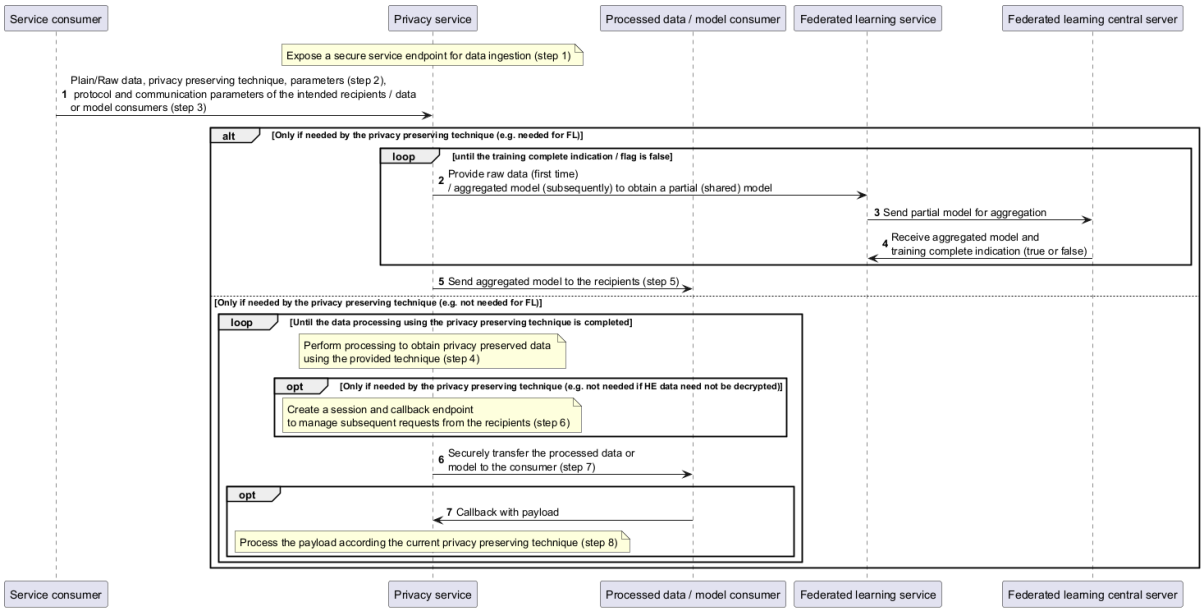


Figure 12: Generic privacy service workflow