Secure Federated Learning in 5G Mobile Networks

Martin Isaksson*‡, Karl Norrman†‡
*Ericsson Research, Artificial Intelligence, Torshamnsgatan 21, SE-164 83, Stockholm, Sweden
†Ericsson Security Research, Torshamnsgatan 21, SE-164 83, Stockholm, Sweden
‡KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science
Email: {martisak, knorrman}@kth.se

***Abstract*—Machine Learning (ML) is an important enabler for optimizing, securing and managing mobile networks. This leads to increased collection and processing of data from network functions, which in turn may increase threats to sensitive end-user information. Consequently, mechanisms to reduce threats to end-user privacy are needed to take full advantage of ML. We seamlessly integrate Federated Learning (FL) into the 3GPP 5G Network Data Analytics (NWDA) architecture, and add a Multi-Party Computation (MPC) protocol for protecting the confidentiality of local updates. We evaluate the protocol and find that it has much lower overhead than previous work, without affecting ML performance.**

***Index Terms*—5G, federated learning, machine learning, security, privacy**

I. Introduction

The 5th generation of mobile network technologies, 5G, defines a new standard for Radio Access Network (RAN) allowing billions of connected devices to transmit more data than ever before. Due to the huge amount of data and devices, complexity of configuring, managing and securing networks increase. To meet these new demands, Machine Learning (ML) is an important enabler [1].

In turn, data collection is necessary for ML. While data collection leads to insights benefiting system optimization, it can be sensitive in a privacy and a business sense, and may be used for nefarious purposes. It is important to respect end users' privacy as well as protecting business sensitive information by considering privacy during the entire network lifecycle [2].

A. Motivation

The telecom industry is now considering collaborative ML to improve privacy when using ML for network optimization, time-series forecasting [3], predictive maintenance and QoE modeling [4], [5].

Collaborative ML such as Federated Learning (FL) proposed for mobile networks in [6], [7] avoids central collection of data and instead perform training of a ML model locally where the data is generated. The local updates generated by, e.g., base stations as in [3], are then aggregated by a parameter server into a new global model. When FL is used, attackers are therefore required to compromise each data generating client individually to obtain its raw data. In this paper we consider precisely these types of 5G use-cases based on a single operator using FL in its own network.

Details of how to realize collaborative learning in 5G on a system architecture and protocol level has not been investigated. Our scheme is therefore relevant for industry and is timely as input for pre-standardization research.

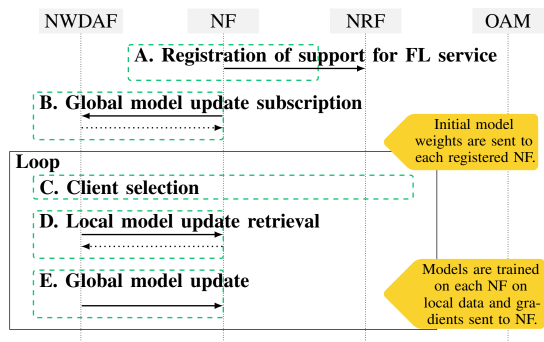

Fig. 1. An overview of our integration of Federated Learning (FL) in a 5G Network Data Analytics (NWDA) context.

B. Assets and Threats

We consider the problem of protecting raw data collected by Network Functions (NFs) in 5G mobile networks for the purpose of tuning their performance. An NF can be deployed at a base station, which is a rich source of data that can be used to infer network-wide insights [1]. Confidentiality of this data could be lost if an NF is compromised. If the data is sent to a central server such as Network Data Analytics Function (NWDAF) for ML model training, there is therefore an increased threat that for example mobility patterns of an end-user leaks. Collecting data also increases the risk that business sensitive information leaks, such as system parameters. Consequently, the central server is a more attractive target because it collects data from multiple NFs.

C. Our main contributions

We design and analyze a scheme which ensures that updates from FL clients are aggregated in a privacy preserving fashion using Multi-Party Computation (MPC). We show how this scheme can be integrated in the 5G Network Data Analytics (NWDA) framework [8], [9] and protocols without breaking the structure of the architecture or its underlying principles. Specifically, we consider the following to be our main contributions:

1. An integration of collaborative learning (in particular Federated Learning (FL)) into the 5G NWDA framework;
2. A privacy-enhancing and efficient scheme for collaborative learning algorithms in the 5G NWDA inspired by [10];

3. An evaluation of the scheme with respect to communication cost, storage cost and computational cost.

II. Background

A. 3GPP 5G Service Based Architecture

The 3GPP created a new framework for core network protocols for 5G called Service Based Architecture (SBA) [11].

1) Architecture and Principles: SBA comprises NFs that expose services through RESTful APIs [12]. NFs can invoke services in other NFs via these APIs. To be discoverable to *service consumer*s, *service producer*s must register with a NF Repository Function (NRF) [13]. Upon request from an NF, the NRF responds with a list of identifiers for suitable service producers, which can fulfill the service criteria posed by the NF. The NF may for example request a list of all service producers of a certain type.

2) Security: SBA builds security in from the start [14] so that access to any API of an NF is authenticated and authorized. Sensitive data transmitted between providers and consumers is further confidentiality and integrity protected.

Operators control a Public Key Infrastructure (PKI) managing certificates for all NFs. Our scheme makes use of this PKI and does not depend on any of the authorization features and we assume that all requests are authorized according to an appropriate policy.

B. 3GPP 5G Analytics framework

For analytics and predictions, 5G provides harmonized mechanisms for data collection. These mechanisms are based on a consumer and producer concept realized by SBA communication patterns and form the framework called NWDA [9]. NWDA is centered around a function named NWDAF that serves two main purposes. First, it acts as a service consumer, collecting data using *Data Collection* procedures from NFs, who act as service providers. Second, it processes the data and provides analytics and predictions as a services provider to other NFs using *Analytics and Prediction Exposure* procedures.

C. Improving privacy with collaborative learning

Use-cases relevant for 5G, listed in Section I-A, encompasses FL tasks such as time-series forecasting for predictive maintenance, or classification for traffic management and QoE modeling. These tasks revolve around a neural network model that is parameterized by weights. We therefore consider neural networks, with weights serialized into a vector $\boldsymbol{w} \in \mathbb{R}^d$, where d is the length of the vector.

In geographically distributed collaborative learning frameworks, the data belongs to and is local to each NF.

The collaborative learning approach FL [6] minimizes objective functions over a set of geographically distributed clients. Training is done synchronously, and a parameter server coordinates the training during a number of training rounds. In training round t, each NF$_k$ will train locally on n_k samples before a local model update $\boldsymbol{w}_{(k,t)}$ is sent to the NWDAF. The NWDAF performs aggregation of all received local updates and distributes the global model to all NFs.

III. Integrating Federated Learning in the Analytics Framework

In this section we present our first main contribution — an integration of collaborative learning, specifically FL, into the 5G NWDA framework.

The NFs trust the NWDAF to faithfully compute the joint model update of the participating NFs individual updates. They would however like to reduce the risk that the NWDAF gets direct access to the raw local data.

The NWDAF may try to insert, delete or modify messages sent between NFs. Because SBA provides confidentiality and integrity protection of messages, NFs or other parties cannot interfere or eavesdrop on the communication between NFs and the NWDAF. This is the trust model we use.

To integrate FL into the NWDA framework, we first split it into five phases, as seen in Fig. 1. We consider the *client* from [6] as a component of an NF and the parameter *server* as a component of a NWDAF.

A. Registration of support for Federated Learning service

An NF may provide the SBA service to train an ML model on local data and send model updates to a subscriber of that service. The model updates are sent as events using the *Data Collection* procedures defined in [9]. We refer to these updates as *local model updates*. The NWDAF is the intended consumer for these services.

During the registration of the ML training SBA service, an NF informs the NRF about supported features related to model training. The NF can for example indicate the current traffic load, whether or not is has a Graphics Processing Unit (GPU) or which types of models it can train.

B. Global model update subscription

An NWDAF provides the NWDA analytics service of sending global model updates to NFs. NFs interested in global model updates need to subscribe to them. It is not required that a consuming NF also makes itself available as a training service provider, but we will assume that this is the case. The number of NFs, K, that have subscribed to the global model updates are available for selection.

C. Client selection

A necessary part of FL is to select a subset of clients in each training round. It is common to use a random subset of all clients. This can be suboptimal in the context of 5G where it can, for example, introduce bias and unfairness [1], [15].

FL algorithms define parameters controlling their behaviors, such as the fraction C of NFs selected for training in each round. We denote the set of selected NFs $\mathcal{L}_\mathrm{S}$ and the number of selected NFs $K_s = |\mathcal{L}_\mathrm{S}| = \lceil CK \rceil$ and note that the selection strategy does not impact the security of our scheme. In each training round, the NWDAF performs client selection by running a *Discovery Request procedure* with the NRF [13]. We enhance client selection in NWDA by allowing the NWDAF to decide the selection strategy and give it means to select NFs by using metadata as indicated in Section III-A or Key Performance Indicators (KPIs) via the NWDA *Data Collection procedure* [9].

D. Local model update retrieval

We integrate sending the local updates with the NWDA analytics subscribe pattern [9]. In this way the NWDAF can trigger the NF to start training of the ML model on local data. When the training is complete, the NF sends the local model update to the NWDAF.

E. Global model update

The NWDAF will aggregate all retrieved local model updates and update the global model. The updated global model is sent to NFs that registered to receive global model updates.

Aggregation of the local model updates is done using a weighted average. The weights depends on the number of local datapoints used in the training round and therefore the local model update also need to contain the number of datapoints.

NFs can stop receiving global model updates from the NWDAF by using the analytics unsubscribe procedure [9], for example when the ML performance of the global model is sufficiently good.

IV. Improving Privacy further using MPC

In this section we present our second main contribution — a privacy-enhancing scheme for FL in the 5G NWDA framework inspired by [10].

A. Residual threats

Even when FL is used, observers may learn sensitive information from the updates themselves. That is, NFs should not reveal their local updates to the NWDAF, nor to other NFs or other parties. Furthermore, FL introduces a new type of sensitive data into the system — the number of datapoints used by each NF. This may leak information about the actual data. Therefore, our scheme also protects the number of datapoints of each NF.

Our scheme ensures that even if NFs collude with the NWDAF to reveal the update of another NF, they will fail.

It is is still possible that some property of the inputs is deducible solely from the output of the computation, i.e., the global ML model [16], [17].

B. Privacy Through Multi-Party Computation

Our protocol consists of two parts, *Session Initialization* and *Aggregation*. The first part establishes a *session* and pair-wise shared secrets between the NFs, and the second executes the aggregation of local updates throughout a number of training rounds. The local updates are protected by masks derived from the pair-wise shared secrets. An example of our scheme with two NFs is depicted in Fig. 2.

1) Preliminaries: Session participants are an NWDAF and a set of NFs we refer to as the total population. The total population is ordered according to some total order determined by the NWDAF, which remains fixed throughout the session. When we refer to the position of an NF, it is w.r.t. this order.

We assume that the PLMN operator maintains a PKI, in which all NFs are enrolled. The NFs are identified by their hostnames, which are unique within the PKI, and their private/public key pair represent their identity. All participants

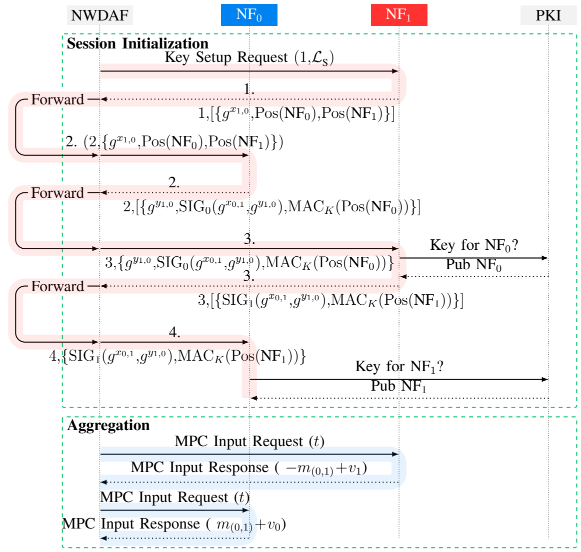

Fig. 2. An overview of Multi-Party Computation (MPC) in Federated Learning (FL) context. In this example only two NFs are selected. The NWDAF instructs NF_1 to initiate a key exchange with all NFs with lower position, in this case only NF_0. Following the session initialization we run an aggregation where the constructed mask $m_{(0,1)}$ is added by NF_0 and subtracted by NF_1. The key K is derived from the Diffie-Hellman secret using a PRF and is part of SIGMA. A list is denoted by brackets, and a container with curly brackets.

have access to a fixed secure PRF and a fixed secure Pseudo Random Generator (PRG) to compute the pair-wise shared secrets and masks, see Fig. 2.

2) Session Initialization: The purpose of session initialization is to establish initialized session states in the NWDAF and all the NFs in the total population. The procedure is point-to-point between the NWDAF and an NF and follows a request/response pattern according to [12, Section 4.6.1]. The procedure tunnels SIGMA key establishment [18] messages between NFs via the NWDAF. SIGMA establishes pair-wise shared secrets between NFs. We allow caching of the pair-wise shared secrets between sessions. A training round sequence number t ensures fresh masks for each training round, even when a pair-wise shared secret from a previous session is re-used. The NWDAF sets t to zero at the start of the session and increases it by one for each round. NFs keep their own local replay counter t_{NF}, and abort if the NWDAF attempts to re-use a lower value for t. The NWDAF sends the first message *Key Setup Request* to all selected NFs, which includes the list $\mathcal{L}_S$.

An NF_k acts as initiator of the SIGMA protocol execution if it has a lower position than another NF. For each NF_i where $i > k$, it creates a container that includes its first SIGMA message.

The NWDAF collects responses from all NFs, before forwarding the containers, in batch, to the correct NF, based on the addresses on the containers. When an NF, say NF_k, receives a list of containers, it creates the corresponding SIGMA response messages, packs them into containers and returns them to the NWDAF. The remaining two SIGMA messages are exchanged similarly.

On completion of SIGMA, the session initialization is considered ready. At this point, all pairs of NFs NF_k and $NF_{k'}$

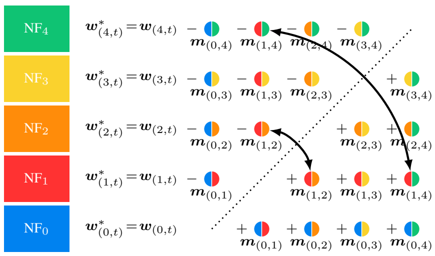

Fig. 3. The 6 selected NFs each add masks for every other NF. The cancellation of masks $\boldsymbol{m}_{(1,2)} =$ and $\boldsymbol{m}_{(1,4)} =$ are indicated by the arrows. Each mask above the diagonal is canceled out by a mask below the diagonal. The cancellations of masks do not affect the aggregated sum, and $\sum_{i=0}^{4} \boldsymbol{w}^*_{(i,t)} = \sum_{i=0}^{4} \boldsymbol{w}_{(i,t)}$.

share a secret $\mathrm{DH}_{k,k'}$ and associate the highest seen training round sequence number with this Diffie-Hellman secret.

3) Aggregation: Local updates are hidden from the NWDAF using masks, which are shared using the secret sharing scheme in Section IV-B2. The masks cancel each other out during the execution of a secure sum protocol, which is an optimization of [19, Protocol 1]. Each NF masks their local updates by independently and randomly sample masks from $\mathbb{Z}_R^d$ for some suitable R, where d is the length of the local update. R must be larger than any component of $\boldsymbol{w}_{(k,t)}$, when the component is interpreted as an unsigned integer. The purpose of R is to constrain the maximum size of components in the masked local update.

The masks are combined with the local update $\boldsymbol{w}_{(k,t)}$ by component-wise integer addition modulo R, where the components are considered as some fixed integer encoding of their respective real-value.

The inverse is the component-wise additive inverse modulo R. Finding the inverse of a mask is trivial for someone who knows the mask and infeasible for anyone else since masks are selected uniformly at random and are independent. Essentially, a masking is an encryption with a one-time pseudo-random pad.

Local updates can be large, therefore, so can the masks. To reduce communication overhead, we use an idea similar to [10], where masks are generated from the pair-wise secret shared between the NF adding the mask and the one canceling it.

Consider masks as vectors of unsigned integers and let the position of a NF be k. NF_k hides its local update $\boldsymbol{w}_{(k,t)}$ by adding all masks $\boldsymbol{m}_{(k,i)}$ to it where $k < i$, and subtracting all masks $\boldsymbol{m}_{(k,i)}$ from it where $k > i$. The masked version $\boldsymbol{w}^*_{(k,t)}$ of the local update is

$$\boldsymbol{w}^*_{(k,t)} = n_k \cdot \boldsymbol{w}_{(k,t)} + \sum_{\substack{i \in \mathcal{L}_{\mathrm{S}}, \\ k < i}} \boldsymbol{m}_{(k,i)} - \sum_{\substack{i \in \mathcal{L}_{\mathrm{S}}, \\ k > i}} \boldsymbol{m}_{(i,k)},$$

where n_k is the number of datapoints for NF_k used in training round t, see Fig. 3 for a visual representation of this.

As a last step, NF_k generates a separate set of masks to hide the number of datapoints used for training. These masks are added and subtracted in the same way as the masks for the local updates, and the NWDAF adds them together and uses the result to scale the sum by the total number of points n used during this training round. This protects the number of datapoints, which would otherwise potentially reveal sensitive information about the local data.

V. Evaluation

In this section we present our third main contribution — an evaluation of the proposed privacy-preserving scheme for FL in 5G NWDA. An argument for the security of the scheme can be found in Appendix A.

A. Setting

Security protocols add overhead, and it is always important to keep the ratio between security overhead and protected data low.

5G datasets can be sensitive in a business sense so real associated models are therefore also unpublished which makes it difficult to obtain and use such a model for our analysis. Therefore, in Fig. 6, as a reading guide for this section we provide a visual representation of the operating point (in terms of number of NFs and model size) for some use-cases. We consider the case where NFs are base stations deployed for one operator and provide the estimated number of NFs based on the number of deployed 4G base stations for a small and a large operator [20].

The size of the ML model depends on the use-case, from very small models in [3] to larger models. As use-cases become more complex, and as the volume of data generated increases, the needed size of models will also increase. We expect the operating point for future 5G use-cases to end up in the green area seen in Fig. 6.

B. Communication cost

We use a communication cost metric where we include the total number of bytes transmitted both in uplink and in downlink. Note that the lower layer protocol overhead, from for example HTTP/2 and TLS, is not included in our analysis.

1) Session initialization: security related communication:

a) First round: In Fig. 2 we see an overview of the messages in the *Session Initialization*. The NWDAF sends a list of hostnames $\mathcal{L}_{\mathrm{S}}$ to each of the selected NFs, K_s times. Each NF will respond to this message with a list of transparent containers, one for each NF that has a lower index than the initiating NF, in total $\binom{K_s}{2}$ such containers. In this way, we calculate the total communication cost of messages in session initialization and compare between the session initialization and aggregation communication cost in Fig. 4.

b) Subsequent training rounds: The probability that selected NFs, NF_i and NF_j, need to exchange secrets in round t is

$$P_{\mathrm{NF}_i \text{ key exchange with } \mathrm{NF}_j} = \left(1 - \frac{K_s}{K}\left[\frac{K_s - 1}{K - 1}\right]\right)^t.$$

As t goes to infinity, this probability goes to 0, as seen in Fig. 5 where the communication cost growth for the session initialization drops off. Note that an NF may already have a certificate of another NF that it obtained for some other reason.

Communication cost in the session initialization is $\mathcal{O}(K_s^2)$. Note that this cost is heavily influenced by the size of the hostname list. Each NF is assigned a 12 B globally unique

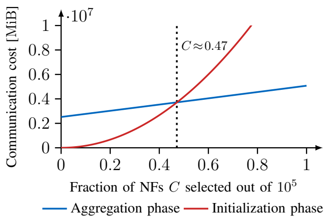

Fig. 4. Comparison of communication costs after one training round between session initialization messages and aggregation messages. We vary the fraction of NFs selected C out of 10^5 NFs. At $C \approx 0.4$ the cost of the session initialization messages exceed the aggregation cost.

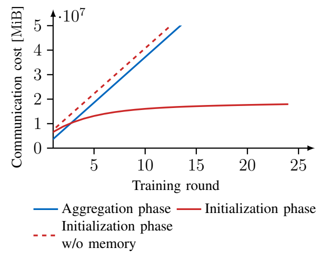

Fig. 5. Comparison of communication costs per training round between session initialization messages and aggregation messages. The number of key establishments are reduced as the number of training rounds increases. The fraction of NFs selected C is from the intersection point in Fig. 4.

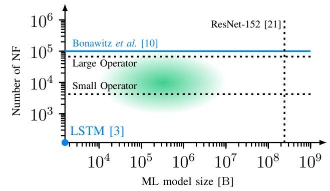

Fig. 6. A guide to understanding the costs and trade-offs we provide ball-park estimates of number of NFs and model sizes for a few use-cases. We expect the operating point for 5G use-cases to end up in the green area.

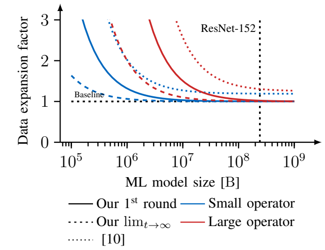

Fig. 7. DEF compared to [10] in the fully malicious case. The line type represents the solution, and the color represents the number of NFs. The decreasing DEF with increasing training rounds of our scheme is shown for the first training round and for the case when all NFs have been selected at least once.

identifier [22]. Adding a domain name, the hostnames we use are 30 B of the form gNB-382A3F47.myran.example.com. Other security parameters are chosen to match 128 bit security.

2) Aggregation: ML model related communication: In each training round each selected NF sends a local model update to the NWDAF, see Fig. 1. The NWDAF performs aggregation and sends the updated global model to all registered NFs. The total cost of sending the model of size d in this step depends on K and on K_s. Each NF has a communication cost of $\mathcal{O}(d)$, so the total aggregation communication cost in each round is $\mathcal{O}(dK)$.

3) Protocol overhead: We compare the total size of all security related messages and model related messages to the protocol without security, i.e. standard FL. We call this overhead DEF. The DEF from security related messages increases with increasing number of NFs. We expect future FL use-cases in 5G to require large models and relatively small number of NFs.

In Fig. 7 we compare the DEF of our scheme to that of [10] in their fully malicious case. This comparison is done for the cases where the number of NFs match the number of base stations from a small and large operators. We vary the model size. As seen in Fig. 5 the DEF for our scheme is reduced with increasing t. In Fig. 7 we plot the DEF for our scheme when all NFs have been selected at least once — there is a similar but much smaller effect for [10] but this is left out from the figure for clarity.

C. Computation cost

1) Session initialization computation: Each NF will as most need to do $K_s - 1$ key establishments in one training round and need to expand the seed to a full mask for every other NF. The mask length depends on the model size. The resulting session initialization computation cost per NF is $\mathcal{O}(K_s d)$.

The session initialization computation cost for the NWDAF depends on the number of selected NFs K_s, and is $\mathcal{O}(K_s)$.

2) Aggregation computation: The NWDAF is unaware of any masks, and simply performs aggregation of K_s local model updates. The resulting session initialization computation cost for the NWDAF is $\mathcal{O}(K_s d)$. The ML computation for an NF is made up of costs for training and inference and is out of scope of this paper.

D. Storage costs

1) Session initialization storage: The largest storage needed is for $K - 1$ session keys, K certificates, K hostnames, 1 private key and 1 training round sequence number. We can trade storage for communication by only storing the certificates and session keys for the NFs that are selected in the current training round. The storage needed in this case then depends on K_s. The storage needed for the NWDAF is for K hostnames and 1 training round sequence number.

2) Aggregation storage: Each participating NF need to store the global model, not counting temporary storage needed during training. The model related storage for each NF is $\mathcal{O}(d)$ and the state related storage is $\mathcal{O}(K_s)$. The model related storage for the NWDAF is $\mathcal{O}(d)$.

VI. Discussion and related work

A. Related Work

Our scheme is inspired by Bonawitz *et al.* [7] in which the authors discuss a practical implementation of FL including security. Their security aspects are further developed in [10]. They target mobile devices with no pre-established security relations and where group membership is volatile. They overcome this volatility by additional functionality in their scheme. However, as we assume that NFs will have a much lower drop rate than mobile phones, we avoid their robustness-improving additions. We also make use of the fact that NFs already are part of a common PKI to reduce complexity. This excludes use-cases with more than one mobile operator, such as [4], [5], and we leave this as future work.

A parameter server may detect malicious or malfunctioning clients based on the information in the local updates. [23] implements a robust Byzantine-resilient aggregation method. Unfortunately, such methods fail when MPC is used, because they need access to the local updates of the FL clients.

[24] proposed to encrypt local updates using Paillier homomorphic cryptosystem which can be more efficient in the initialization phase, but they don't evaluate this. They also show that parameter server complexity increases, and that ML

performance is lower. They do not consider how to embed their protocol in any particular system.

Although MPC is applied, the global model may leak information. To cope with that differential privacy could be applied, but those schemes need further work in before they can be practically applied [15].

VII. Conclusion

ML is becoming an essential technology for optimizing mobile networks. This has lead to an increased collection and processing of data that may leak sensitive information. Consequently, mechanisms to protect the business sensitive information and end-users' privacy are needed.

We devised a scheme for end-user privacy protection and demonstrated how to integrate it in the 5G SBA and NWDA architecture. The scheme was evaluated in terms of computational and communication cost. We explore the security of our scheme in Appendix A.

We found that the DEF depends on the client fraction C, the size of the ML model, the number of NFs and the training round t. For the use-cases we envision, as well as for potential future use-cases we showed that the overhead of our scheme is smaller than that of [10]. Our gain stemmed from relaxed reliability constraints and re-use of existing telecom infrastructure, such as PKI. However, we see an opportunity to further improve our scheme in terms of communication overhead and to use our NWDA integration to improve bias and fairness.

Even though it is known that sensitive information may still leak even when FL and MPC are properly applied, our scheme significantly improves privacy. Because it is available and much simpler to apply in practice, in comparison to differential privacy, we believe it would be beneficial to deploy a scheme such as ours.

Acknowledgment

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

We would especially like to thank Prof. Patric Jensfelt, Prof. Mads Dam and Dr. Rickard Cöster.

References

[1] E. Fersman, J. Forgeat *et al.*, "Artificial intelligence and machine learning in next-generation systems," Ericsson Research, Ericsson AB, Tech. Rep., 2018. [Online]. Available: https://www.ericsson.com/en/white-papers/machine-intelligence

[2] M. Anneroth, D. Casella *et al.*, "Privacy in mobile networks — How to embrace privacy by design," Ericsson Research, Ericsson AB, Tech. Rep., 2019. [Online]. Available: https://www.ericsson.com/en/white-papers/privacy-in-mobile-networks

[3] F. Díaz González, "Federated Learning for Time Series Forecasting Using LSTM Networks: Exploiting Similarities Through Clustering," Master's thesis, KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science, 2019. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-254665

[4] S. Ickin, K. Vandikas, and M. Fiedler, "Privacy preserving QoE modeling using collaborative learning," *4th Internet-QoE Workshop: QoE-based Analysis and Management of Data Commun. Networks (Internet-QoE 2019)*, Oct. 2019. [Online]. Available: https://doi.org/10.1145/3349611.3355548

[5] K. Vandikas, S. Ickin *et al.*, "Privacy-aware machine learning with low network footprint," Ericsson Research, Ericsson AB, Tech. Rep., 2019. [Online]. Available: https://www.ericsson.com/en/ericsson-technology-review/archive/2019/privacy-aware-machine-learning

[6] B. McMahan, E. Moore *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Proc. of the 20th Int. Conf. on Artificial Intelligence and Stat.*, ser. Proc. of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, Apr. 2017, pp. 1273–1282. [Online]. Available: http://proceedings.mlr.press/v54/mcmahan17a.html

[7] K. Bonawitz, H. Eichner *et al.*, "Towards federated learning at scale: System design," *CoRR*, vol. abs/1902.01046, 2019. [Online]. Available: http://arxiv.org/abs/1902.01046

[8] 3GPP, "System architecture for the 5G System (5GS)," TS 23.501, Apr. 2019, version 16.0.2. [Online]. Available: http://www.3gpp.org/DynaReport/23501.htm

[9] 3GPP, "Architecture enhancements for 5G System (5GS) to support network data analytics services," TS 23.288, Jun. 2019, version 16.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/23288.htm

[10] K. Bonawitz, V. Ivanov *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. of the 2017 ACM SIGSAC Conf. on Comput. and Commun. Security*, ser. CCS '17. New York, NY, USA: ACM Press, 2017, pp. 1175–1191. [Online]. Available: http://doi.acm.org/10.1145/3133956.3133982

[11] 3GPP, "Technical Realization of Service Based Architecture," TS 29.500, Jun. 2019, version 16.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/29500.htm

[12] 3GPP, "5G System; Principles and Guidelines for Services Definition; Stage 3," TS 29.501, Jun. 2020, version 16.3.0. [Online]. Available: http://www.3gpp.org/DynaReport/29501.htm

[13] 3GPP, "5G System; Network function repository services; Stage 3," TS 29.510, Mar. 2020, version 16.3.0. [Online]. Available: http://www.3gpp.org/DynaReport/29510.htm

[14] 3GPP, "Security architecture and procedures for 5G system," TS 33.501, Sept. 2019, version 16.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/33501.htm

[15] P. Kairouz, H. B. McMahan *et al.*, "Advances and open problems in federated learning," *CoRR*, vol. abs/1912.04977, 2019. [Online]. Available: http://arxiv.org/abs/1912.04977

[16] B. Hitaj, G. Ateniese, and F. Pérez-Cruz, "Deep models under the GAN: information leakage from collaborative deep learning," in *Proc. of the 2017 ACM SIGSAC Conf. on Comput. and Commun. Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, 2017, pp. 603–618. [Online]. Available: https://doi.org/10.1145/3133956.3134012

[17] Z. Li, Z. Huang *et al.*, "Quantification of the Leakage in Federated Learning," in *Workshop on Federated Learning for Data Privacy and Confidentiality (in Conjunction with NeurIPS 2019)*, vol. abs/1910.05467, 2019. [Online]. Available: http://arxiv.org/abs/1910.05467

[18] H. Krawczyk, "SIGMA: the 'sign-and-mac' approach to authenticated diffie-hellman and its use in the ike-protocols," in *Advances in Cryptology - CRYPTO 2003, 23rd Annu. Int. Cryptology Conf., Santa Barbara, California, USA, August 17-21, 2003, Proc.*, ser. Lecture Notes in Computer Science, D. Boneh, Ed., vol. 2729. Springer, 2003, pp. 400–425. [Online]. Available: https://doi.org/10.1007/978-3-540-45146-4_24

[19] G. Kreitz, M. Dam, and D. Wikström, "Practical private information aggregation in large networks," in *Information Security Technology for Applications - 15th Nordic Conference on Secure IT Systems, NordSec 2010, Espoo, Finland, October 27-29, 2010, Revised Selected Papers*, ser. Lecture Notes in Computer Science, T. Aura, K. Järvinen, and K. Nyberg, Eds., vol. 7127. Springer, 2010, pp. 89–103. [Online]. Available: https://doi.org/10.1007/978-3-642-27937-9_7

[20] "Cellmapper." [Online]. Available: https://www.cellmapper.net/networks

[21] K. He, X. Zhang *et al.*, "Deep residual learning for image recognition," in *2016 IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[22] 3GPP, "Common Data Types for Service Based Interfaces," 3GPP, TS 29.571, Jun. 2020, version 16.3.0. [Online]. Available: http://www.3gpp.org/DynaReport/29571.htm

[23] G. Damaskinos, E. Mahdi *et al.*, "AGGREGATHOR: Byzantine Machine Learning via Robust Gradient Aggregation," *Proc. of the 1st SysML Conf.*, 2019. [Online]. Available: https://www.sysml.cc/doc/2019/54.pdf

[24] J. Zhang, B. Chen *et al.*, "PEFL: A privacy-enhanced federated learning scheme for big data analytics," in *2019 IEEE Global Commun. Conf., GLOBECOM 2019, Waikoloa, HI, USA, December 9-13, 2019*. IEEE, 2019, pp. 1–6. [Online]. Available: https://doi.org/10.1109/GLOBECOM38437.2019.9014272

Appendix

A. Security justification

Although we have explained the security purpose for introducing functionality throughout the paper, we now give a brief argument for the security of the compound scheme. The correctness of the protocol can be seen from the description of the protocol itself above, and we will not consider it further.

The security goal of the scheme is to ensure that the NWDAF only knows its own input to the computation and the final result, i.e., the updated model. The initialization phase is run once and the masked-based secure sum protocol is run each round to compute a new updated model. Our argument that this is secure can be divided into the following two claims.

Claim 1: The mask-based secure sum protocol fulfills the security goal assuming the masks are uniformly and randomly selected and assuming each pair of NFs share a mask and its inverse (we call these two pair-wise masks below), known only to that NF-pair.

Claim 2: The masks are uniformly and randomly selected in each round assuming SIGMA is a secure key establishment protocol, that the PRF, PRG, signature scheme and MAC are secure according to standard definitions.

To justify *Claim 1* we argue as follows. For each other NF, an NF adds the mask it shares with that NF to the local update (or the inverse of the mask depending on their relative positions in the NF order). Adding a uniformly random mask to a local update using modular addition results in a uniform distribution. No-one except that pair of NFs can hence distinguish the masked local update from a random value. For each pair of NFs, the NF adds a mask known only to that pair. This means that given the total sum of all the masks and the local update contains at least $|\mathcal{L}_{\mathrm{S}}|-2$ masks not known to any given NF. Consequently, at least $|\mathcal{L}_{\mathrm{S}}|-2$ need to collude to unmask a masked local update. Adding two masked local updates together will provide the sum of those local updates still masked by the remaining masks. As long as at least one mask remains in the sum, the sum cannot be unconcealed. Once all local updates are added together, the result is their sum and all masks are canceled. At this point the NWDAF knows the output of the computation, but has not been able to unconceal any of the inputs, which is what we claimed.

To justify *Claim 2* we argue as follows. In the initialization phase, each NF establishes a pair-wise secret with each other NF using the SIGMA key establishment protocol. Further, the signature scheme, MAC and MAC-key generation via the PRF, on which SIGMA relies, are secure by assumption. SIGMA is secure in the CK-model [18], meaning that the established shared secret is indistinguishable from a randomly selected element from the underlying Diffie-Hellman group. We can therefore assume that the pair-wise shared secret is indistinguishable from random to anyone else than the pair of NFs and that it is mutually authenticated.

In each round, each NF verifies that t has not been used earlier, and we therefore can assume it is fresh for all NFs in all runs of the protocol. For simplicity, we assume that an NF that detects a re-used t value stops execution, at which point the entire round of the protocol fails to execute. Note that even in that case, only NFs which obtain a fresh t value would continue execution, so all NFs can be assumed to use a fresh t value to generate output in the protocol.

The pair-wise masks are generated from the pair-wise shared secret, which may be the same for more than one training round. However, the value t, which is guaranteed to be fresh for each training round, is also used as input to the mask-generating PRG. Because the PRG is secure, its output is indistinguishable from a uniformly randomly selected string given that the input g^{xy} obtained from SIGMA is secret. To conclude, because t is fresh, the pair-wise masks are uniformly random, known only to the NF-pair, and they are secret and fresh for each round.

We note that the order of the NFs affects two aspects. First, the order determines which NF acts as initiator and which one acts as responder for the SIGMA exchange between each pair. SIGMA is secure regardless which part takes which role, and no NF will continue execution of the scheme unless it has run SIGMA with each NF in the order. So, the NWDAF does not gain anything by selecting a certain order in this respect.

Second, the order determines which NF computes a mask and which NF computes the inverse of a mask. By the symmetry of the masks, it is irrelevant which NF generates the mask. Since no NF will continue execution unless it has a pair-wise mask with all other NFs in the pair, the NWDAF gain nothing by selecting a certain order.

The secure sum protocol is considered meeting the security goal even in the degenerate case where the $|\mathcal{L}_{\mathrm{S}}|=1$. In that case, the NWDAF would in fact learn the local update of the single participating NF. This can be prevented by adding a rule in the scheme that NFs shall terminate the execution if the size $\mathcal{L}_{\mathrm{S}}$ is less than some threshold value. It may be useful to set this threshold to a larger value than one to reduce the effects of outliers.