Future Generation Computer Systems 152 (2024) 288–303

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Anonymous Federated Learning via Named-Data Networking

Andrea Agiollo [a,1], Enkeleda Bardhi [b,*,1], Mauro Conti [c], Nicolò Dal Fabbro [d,1], Riccardo Lazzeretti [b]

[a] Department of Computer Science and Engineering, University of Bologna, Italy
[b] Department of Computer, Control and Management Engineering, Sapienza University of Rome, Italy
[c] Department of Mathematics, University of Padua, Italy
[d] Department of Information Engineering, University of Padua, Italy

ARTICLE INFO

Keywords:
Federated learning
Anonymity
Named data networking
Privacy-preserving protocols
Next generation communication

ABSTRACT

Federated Learning (FL) represents the de facto approach for distributed training of machine learning models. Nevertheless, researchers have identified several security and privacy FL issues. Among these, the lack of anonymity exposes FL to linkability attacks, representing a risk for model alteration and worker impersonation, where adversaries can explicitly select the attack target, knowing its identity. Named-Data Networking (NDN) is a novel networking paradigm that decouples the data from its location, anonymising the users. NDN embodies a suitable solution to ensure workers' privacy in FL, thus fixing the abovementioned issues. However, several issues must be addressed to fit FL logic in NDN semantics, such as missing push-based communication in NDN and anonymous NDN naming convention. To this end, this paper contributes a novel anonymous-by-design FL framework with a customised communication protocol leveraging NDN. The proposed communication scheme encompasses an ad-hoc FL-oriented naming convention and anonymity-driven forwarding and enrollment procedures. The anonymity and privacy requirements considered during the framework definition are fully satisfied through a detailed analysis of the framework's robustness. Moreover, we compare the proposed mechanism and state-of-the-art anonymity solutions, focusing on the communication efficiency perspective. The simulation results show latency and training time improvements up to ∼30%, especially when dealing with large models, numerous federations, and complex networks.

1. Introduction

Federated Learning (FL) [1] represents the most popular and highly adopted framework for enabling multi-party joint training of Machine Learning (ML) and Deep Learning (DL) models. Most, if not all, FL scenarios consider a central server – namely Central Aggregator (CA) – interacting with multiple users – also called clients or workers – to train a ML model jointly. In this setup, each worker locally trains its ML model on its private data, while the CA aggregates local updates upon their reception. The groundbreaking idea behind FL is that the training process of the global model takes into account the data of all clients belonging to the federation while never disclosing their nature. Indeed, each worker never shares local raw data but propagates updates from the training procedure over such data. Moreover, FL achieves efficiency improvements over centralised training approaches, thanks to its underlying parallelisation paradigm.

Despite the data privacy design of FL, many state-of-the-art works show that local updates shared in the federation process may leak user information. Thus, private knowledge concerning the user data or its identity can still be leaked by local updates and global aggregation—either under the form of ML model [2], or its gradient [3–6]. Here, the protection of user local information is twofold and follows the anonymity vs. privacy trade-off [7]. More in detail, when considering protecting users from information leakage, it is required to *(i)* protect the value of local models from leaking, and *(ii)* hide clients' identities in local parameter updates. Most, if not all, of the effort in the research community has focused on *(i)*, where the concept of Differential Privacy (DP) has been widely used to ensure the protection of local parameters [8]. Here, noisy data are injected into local updates to avoid disclosure of information [9,10].

Although crucial, data privacy techniques focus solely on the data perspective, ignoring issues arising from client identity leakage. Here, linkability attacks [11] represent a critical issue in FL applications. An adversary model comprises curious or malicious CA or a set of clients interested in disturbing the local model of a particular client.

[*] Corresponding author.
E-mail address: bardhi@diag.uniroma1.it (E. Bardhi).

[1] Co-first authors.

https://doi.org/10.1016/j.future.2023.11.009
Received 1 May 2023; Received in revised form 3 November 2023; Accepted 7 November 2023
Available online 10 November 2023

An additional adversary model includes an eavesdropper or a man-in-the-middle located inside or outside the federation aiming to tap or disrupt the communication between the CA and a particular client, respectively. To achieve their aim, the adversaries must know the client's identity, i.e., IP address, so the altered model parameters are sent only to the targeted client. These issues are particularly relevant in the context of FL scenarios where competitor parties or public organisations represent the federation actors. For example, consider the following scenarios: *(i)* the health-care domain solutions where multiple hospitals aim at jointly optimising a target ML model [12]; *(ii)* the Vertical Federated Learning (VFL) paradigms, requiring collaboration between competitor companies [13]; and *(iii)* the smart city scenario, where devices and data are characterised by high heterogeneity [14]. Here, the anonymity of collaborating competitors must be ensured to avoid eavesdropping, man-in-the-middle, and other threats.

Anonymous communication represents a well-known paradigm investigated to achieve privacy-preserving solutions in FL [15–19]. Generally, these approaches leverage known tools for anonymous communication over the IP-based Internet, e.g., Tor [20]. Although valid, such solutions introduce performance issues mainly due to the computation overhead while also being vulnerable to deanonymisation via traffic analysis attacks [21–23].

In this paper, we propose a novel anonymity framework by leveraging the Information-Centric Networking (ICN) paradigm for ensuring client anonymity in FL applications. First introduced in the TRIAD project [24], ICN aims at replacing the IP layer of the Internet with a novel content-centric layer. Among the variety of ICN architectures that have been proposed, Named-Data Networking (NDN) [25,26] is considered the most promising. NDN removes IP addressing and refers to the data using application-level names. In particular, NDN users look for content by name via interest requests. Removing the IP addressing procedure, NDN is considered a privacy-preserving and anonymous-by-design paradigm.

Integrating FL upon an underlying NDN network is challenging. Indeed, the contrast between the pull-based nature of communication in vanilla NDN and the push-based approach characterising every FL framework calls for a careful protocol design. Additionally, anonymous FL requires a customised trust scheme that hides the identities and avoids linkability attacks. Some efforts have been put into overcoming the former limitations, mainly focusing on the interest expiration time to enable dynamic content retrieval and distributed computing [27,28]. Nevertheless, these approaches are unsuitable for our anonymous FL scenario, as they do not enable the clients to act as consumers, i.e., receiving global model, and producers, i.e., pushing the local model to the CA. Instead, for the latter limitation, state-of-the-art vanilla NDN solutions [29,30] bind namespaces to producer identities, thus failing at ensuring full anonymity for FL workers. Therefore, we analyse how to integrate FL into an NDN networking scenario, enabling bidirectional communication and worker anonymity. While avoiding introducing requirements on local updates, our mechanism can be coupled with any data-privacy approach, thus representing the first – up to our knowledge – protocol that achieves complete anonymity in FL. We then analyse the proposed scheme's anonymity in detail, considering different attacks to disclose users' information. Finally, we study latency performance improvements arising from the introduction of NDN communication, primarily deriving from the in-network caching paradigm characterising NDN.

Contributions: We summarise our contributions as follows:

- We present the first NDN-based framework for anonymous FL. The proposed scheme is designed to achieve client anonymity and avoid identity disclosure to other nodes inside or outside the federation network.
- We analyse the anonymity of the proposed approach over a set of well-defined threats in FL. The proposed approach shows reliability against identity disclosure.
- We study the performance of the proposed framework for deploying FL scenarios. The proposed communication scheme is compared against traditional IP-based FL and other available anonymous communication frameworks, showing improved performance over the state-of-the-art.

Table 1
Summary of notations.

Symbols	Definition
N	Number of workers
ID_{an}	Anonymous worker ID
W_i or $W_{ID_{an}}$	ith or ID_{an}^{th} worker
$M^{(t)}$	Global model at time t
$m_i^{(t)}$ or $m_{ID_{an}}^{(t)}$	ith or ID_{an}^{th} local model at time t
$s_{ID_{an}}^t$	Secret factor for ID_{an} at time t
$\lambda_{ID_{an}}$	Secret parameter between ID_{an} and CA
$h(\cdot)$	Hash function
$k_{ID_{an}}$	Symmetric key between CA and $W_{ID_{an}}$
$\mathcal{E}_s(\cdot)$	Symmetric encryption function
$\mathcal{D}_s(\cdot)$	Symmetric decryption function
k_{CA}^{priv}	CA's private key
k_{CA}^{pub}	CA's public key
$\mathcal{E}_a(\cdot)$	Asymmetric encryption function
$\mathcal{D}_a(\cdot)$	Asymmetric decryption function
$chal_i$	ith challenge
$\mathcal{F}_{puf}(\cdot)$	PUF function
res_i	ith response
$tab\langle chal, res\rangle$	Table with challenges and responses
ϵ	Security factor
n_i^{hops}	Number of hops between W_i and CA
B^{hop}	Bit rate of given hop
D^{hop}	Delay of given hop
T^{hop}	Latency of given hop
P_S	Packet size
S	Model size

Organization: Section 2 discusses the basic concepts of FL and NDN, while the related work is presented in Section 3. Section 4 describes the system model and the requirements to be satisfied while designing our mechanism. Section 5 presents the details of our proposed framework, while Section 6 discusses how and to what extent user anonymity is achieved. Our protocol's performance is detailed in Section 7. Finally, Section 8 provides conclusions and insight into possible extensions of our work.

Notations: Table 1 summarises notations used in the article.

2. Background

In this section, we briefly provide a background on the two pillars of our work, i.e., FL in Section 2.1 and NDN in Section 2.2.

2.1. Federated learning

Most, if not all, FL scenarios consider a central server – namely CA – interacting with multiple clients to train a ML model jointly. In this setup, we consider a federation network made of several *clients* – which we refer to also as FL *workers* or *users* –, a CA agent, and the router devices interconnecting the parties. Each worker locally trains its ML model on its private data for a predefined time. Thus, depending on its available data, each federation client obtains a local update over the global ML model. The FL training relies on clients periodically sending the results of their local training – under the form of whole model [31,32], obtained gradient [33,34], or other possible solutions – to the CA agent. The CA is then in charge of aggregating the local updates received by the federation clients to compute the joint global update. Different aggregation solutions have been proposed recently, such as FedAvg [33] or Newton-type methods [35]. Once the global update is available, the CA propagates it to all the federation participants, ensuring that each worker is constantly synchronised with the global state of the federation. The global optimisation process is repeated T times to achieve model convergence.

2.2. Named data networking (NDN)

NDN [36] is a new networking paradigm proposed among five projects of the U.S. National Science Foundation under the Information-Centric Networking (ICN) architectures [37]. NDN represents a data-centric network architecture, where the data is explicitly named and referred to during the communication. The communication in NDN follows a pull-based fashion, i.e., consumers request content through interest packets, while the producers provide the data upon request via the data packets. In particular, NDN introduces hierarchically structured unique names – similar to Uniform Resource Locators (URLs) – for requesting and retrieving the data. For example, a consumer interested in watching the Netflix series X, episode 1, issues an interest packet containing the name */Netflix/series/X/ep/1*. When the request reaches the producer, a data packet with the same name delivers the requested data to the consumer. To embody this communication shift, NDN modifies the routing and forwarding engines to enable the name parsing for delivering the packets to the destination. Here, NDN enables caches on the routers, where the data is stored for later use according to a caching policy, improving the retrieval latency. Additionally, the routers keep track of the requests for content that cannot be satisfied locally based on their arrival interface in the Pending Interest Table (PIT) and use the reverse-path routing to deliver the data to the consumer once it is retrieved from the producer. The router consults its Forwarding Information Base (FIB) table whenever the content request is not in the PIT. Decoupling the content from its location – i.e., avoiding using IP addresses – NDN ensures complete anonymity for consumers. Furthermore, NDN demands security guarantees encapsulated in the data packet. The content producers sign the data packets to ensure data origin authentication and integrity.

Although large-scale NDN networks are not yet available, the research community has shown the benefits of this paradigm in the most disparate scenarios, i.e., Internet of Things (IoT) [38–40], Vehicular Networks [41,42] and Blockchain [43]. Furthermore, research has investigated the integration of NDN paradigm into today's infrastructure, i.e., TCP/IP, proposing various solutions of placing NDN semantics over or under the IP protocol [44]. Given the benefits of applying NDN to several applications, we aim to transpose them to the FL scenarios in this paper.

3. Related work

This section provides an overview of the research effort for providing privacy-preserving techniques for FL and NDN.

Privacy in FL. To achieve privacy-preserving FL, the following techniques can be used: *(i)* Differential Privacy (DP), *(ii)* blockchain-based and *(iii)* communication anonymity. Although our mechanism falls in the communication anonymity group, we present the state-of-the-art for all three groups. DP techniques, i.e., *(i)* focus on the privacy of the data by adding noise to personal sensitive attributes [45]. Zhu and Yu [46] elaborate on the DP benefits and definitions for different scenarios, also among distributed ML. Other works [47,48] engage secure multiparty computation and DP to achieve a secured FL model with high accuracy. Since DP is the most popular approach to ensuring privacy in FL, several works focus on proposing various DP solutions over different FL setups. Hu et al. [49] and Wei et al. [50] focus on personalised FL, highlighting the complexity of keeping privacy while also needing to customised models depending on users needs. Several other works focus on the communication aspects impacting DP, focusing on different types of networks, such as wireless networks [51], Internet of Vehicles [52], and many more. Finally, several works analyse the possible impacts of DP on different FL application scenarios, e.g., healthcare [53], mobility solutions [52]. Interested readers can refer to recent surveys [54,55] elaborating more on the DP techniques.

Among the blockchain-based technologies, i.e., *(ii)*, Shayan et al. [56] tackle privacy issues arising from a single point of trust in FL scenarios, e.g., CA, by leveraging blockchain to certify model updates in a peer-to-peer fashion. Miao et al. [57] propose leveraging blockchain technologies to define federations robust against byzantine attacks. Similarly, BlockFL [58] focuses on possible malicious nodes, adding a consensus process to validate local models effectiveness. Several works also propose different blockchain-based technologies depending on the application scenario at hand, such as healthcare [59], industry [60] and Internet of Things (IoT) [61]. Other popular approaches have been recently proposed, leveraging blockchain technologies to ensure privacy requirements in domains such as IoT [62,63]. Although not focusing specifically on FL, these approaches could be transferred to the FL domain with limited effort. We refer interested readers to recent studies [64–66] providing surveys of the blockchain-based FL frameworks used to boost the FL security and privacy issues.

Among communication anonymity techniques, i.e., *(iii)*, FedTor [15] is a recent anonymity framework proposed for IoT-based FL that leverages the onion router (TOR) to provide worker anonymity.

Domingo et al. [16] proposed a FL framework that offers both security and privacy. Their framework aims to achieve the unlinkability between worker identity and its updates. To achieve this, the workers forward the updates to other workers before sending them to the central aggregator. Additionally, Li et al. [17] targeted the adoption of FL in autonomous driving and proposed an anonymous privacy-preserving scheme. This scheme mainly focuses on how to keep the worker identity private and realises it by adopting zero-knowledge proofs (ZKP). Girgis et al. [19] proposed a privacy amplification for the FL considering sampling and shuffling on data and clients, i.e., anonymisation. In such a way, it enables the transferability of local privacy to the central privacy guarantees. Lyu et al. [18] introduced a fair and privacy-preserving deep learning (FPPDL) framework that designs a local credibility mutual evaluation mechanism to guarantee fairness. Additionally, it uses a three-layer onion-style encryption scheme to guarantee the accuracy and privacy of the individual model updates.

Privacy in NDN. Efforts in privacy-preserving techniques in NDN mainly include anonymity solutions [67,68]. ANDaNa [67] aims to achieve communication anonymity by borrowing several features from Tor. Similarly, Kita et al. [68] focus on the Tor logic to achieve content-producer unlinkability. Furthermore, NDN-ABS [69] poses on the producer anonymity issue of NDN and proposes a novel signature based on attributes. Although valid, the proposed anonymity solutions for NDN are insufficient for the FL scenarios where clients act as consumers and producers. In this respect, the FL scenarios expect modifications on the communication paradigm to enable the clients to push information towards CA. Restricted research on NDN [27,28] aim at modifying the interest-data exchange in vanilla NDN for scenarios of dynamic data retrieval and distributed computing, respectively.

Similarly, few works focus on defining publish/subscribe infrastructures in the NDN domain by modifying the vanilla-NDN workflow [70,71]. While beneficial from the communication perspective, these works lack an anonymity-enabling scheme, representing an essential requirement in FL scenarios. Therefore, an ad-hoc mechanism that satisfies anonymity and communication schemes is required for FL

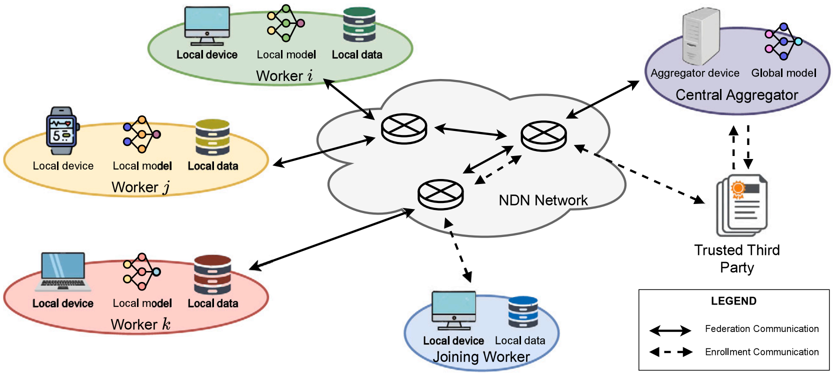

Fig. 1. System model of the proposed framework, including FL workflow and worker enrollment procedure.

4. Federated learning via named data networking

Federated Learning (FL) encounters several privacy issues; among these, anonymity is not yet thoroughly explored. Indeed, most of this research focuses on data privacy [46–48] while ignoring privacy issues emerging from revealing a worker's identity. Only recently, the communication anonymity aspect in FL has gained attention in the research community, with few preliminary works in different domains [15–17]. Furthermore, recent research advancements argue the importance of networking design in distributed learning tasks, like FL [72–74]. In light of such progress, focusing on the FL anonymity perspective from the networking point of view represents a relevant research opportunity for further FL advancements. Given its anonymity by design feature, in this context, the Named-Data Networking (NDN) paradigm represents an appealing choice to develop anonymous FL routing protocols. Although NDN has been proposed as a novel networking paradigm to replace the IP layer, the research showed that it could also be implemented on top of the IP protocol itself [75–77]. Therefore, we claim the NDN-based routing protocol to be a general tool that can be applied to FL scenarios in the current Internet infrastructure. We describe our system model in Section 4.1, threat model in Section 4.2, and elaborate on the system requirements in Section 4.3.

4.1. System model

Our mechanism encompasses a native NDN network, a Central Aggregator (CA) server, and a Trusted Third Party (TTP). The former comprises N FL workers, denoted as W_i, $i \in [1, N]$. Instead, the CA server orchestrates the global model training by synchronising and aggregating the local updates to compute a global update which is then distributed back to the workers. Lastly, the TTP is in charge of verifying and enrolling the workers in the federation. Similarly to existing privacy-preserving solutions for FL [78–80], we consider TTP to be a trusted entity that holds information related to the workers and handles their anonymous enrollment in the federation. Fig. 1 shows an example of the proposed system architecture.

The proposed system incorporates two phases: *(i)* enrollment, and *(ii)* training phase. For the *enrollment* phase, we consider a lightweight anonymity-preserving scheme where the worker responds to the challenges issued by TTP entity. Our mechanism relies on the Physically Unclonable Functions (PUFs) [81–83] to preserve the worker's anonymity. PUFs are hardware-specific elements considered as the unique physical identity of the devices widely leveraged in the cryptography field. After that, the anonymous identity embedding is communicated to the CA server by the TTP. Finally, after further verifying the worker, the CA picks the secret cryptographic parameters and shares them with the worker. For the *training* phase, the proposed framework encompasses different training rounds, each of which composed by the traditional four steps of FL training [1], namely:

1. *Global update*. Here, the CA transmits the global update to each client in the federation.
2. *Local update*. The federation clients optimise their model parameters based on their training dataset.
3. *Update transmission*. Here, the model parameters obtained by each worker are shared with the CA through the anonymous channel.
4. *Model aggregation*. The CA aggregates all the local updates, computing the global model for the next round.

For the remainder of the paper, for simplicity, we denote the exchanged local and global updates by *model* parameters, i.e., weights. Without any loss of generality, the proposed approach can be applied to any model-related updates, e.g., gradient [33,34], as our mechanism solely focuses on the communication perspective. Therefore, in our framework, we refer to the ith worker with the notation W_i and to its corresponding real (anonymous) identity as ID_i (ID_{an}). The local model obtained by the worker having identity ID_{an} at the tth training round is indexed as $m_{ID_{an}}^{(t)}$. Meanwhile, the global model aggregation is defined mathematically as:

$$M^{(t)} = \frac{1}{N} \sum_{i \in \mathcal{F}} m_i^{(t)}. \tag{1}$$

where, N represents the total number of clients in the federation $\mathcal{F}$ and $M^{(t)}$ identifies the global model computed at round t.

4.2. Threat model

The considered threat model encompasses adversaries that behave as *(i)* a single or a set of malicious workers placed inside or outside the federation, *(ii)* a malicious CA, or *(iii)* a honest but curious CA. Considering the above threats, we further elaborate on the adversaries' capabilities in each case.

- *Malicious Workers.* A malicious entity inside or outside the federation aims to eavesdrop on the links connecting the targeted worker with the CA. Eventually, having enabled eavesdropping, such malicious workers could perform a man-in-the-middle attack To this end, the adversary must know the targeted worker's identity, e.g., the IP address We define these threats as *in federation eavesdropping* and *out of federation eavesdropping*, respectively Additionally, malicious workers capable of eavesdropping on other workers' identities, e.g., IP addresses or similar, can pose as them by identity spoofing We define this threat as *worker impersonation* In such a way, an attacker can send corrupted updates, linking them with an honest user or reconstructing the target's data by observing consecutive updates to achieve privacy leakage
- *Honest but curious CA.*
  In this scenario, the CA server executes aggregation computation correctly and provides the worker with the correct parameters.

However, the CA is curious and aims at inferring information concerning the local private data of a single or a set of workers from the received updates

- *Malicious CA.*
  Similarly to the previous scenario, the CA targets a single or a set of workers aiming at corrupting their local models. Here, the CA exploits the knowledge of workers' identities, e.g., IP address, to alter the local model by forwarding perturbed updates to the targeted worker.

4.3. System requirements

To design a robust anonymous FL framework, we establish the following requirements that must be addressed:

R1 *Worker's identity anonymity.*
The fundamental characteristic of our framework is to keep the identity of the clients completely anonymous, even to the CA This requirement represents a fundamental specification to achieve a two-fold privacy-preserving FL scenario

R2 *Robust against misbehaving parties.*
Common FL frameworks are susceptible to misbehaving workers inside and outside the federation who exploit identity spoofing to impersonate workers participating in the federation Similarly, such frameworks suffer from malicious and honest but curious CA server behaviours that perturb the local models and reconstruct the local data

R3 *Compatibility with data privacy mechanisms.*
While designing the anonymity scheme for FL framework, we require it to be compatible with other data-focused privacy mechanisms such as DP Indeed, the privacy issue in FL is two-folded, i.e., ensuring both data privacy and user anonymity. Thus, it would be deleterious to ensure only user anonymity (data privacy) while disregarding data privacy (user anonymity)

Our proposed approach, presented in detail in Section 5, aims at incorporating all the requirements R1-R3. In particular, we use NDN networking principle to enable anonymous communication between the CA and each worker and rely on the TTP during clients' enrollment to ensure their anonymity even to the CA, thus satisfying R1. Meanwhile, to achieve R2, we design a novel naming schema for the underlying NDN network to protect the communication from malicious workers, aiming at impersonating other workers or eavesdropping on their identity, see Section 5.1. We also propose to modify the standard caching mechanism that characterises NDN, presenting a novel caching paradigm. Such a modification, along with the leveraged enrollment scheme, allows for achieving protection of the federation from malicious CA server. Finally, leveraging the NDN paradigm, our proposed approach focuses solely on the communication perspective, avoiding introducing requirements concerning the type of updates forwarded or their structure. Therefore, our mechanism is entirely independent of any data-privacy approach and can be easily integrated with any of them, e.g., DP techniques, satisfying R3.

5. Proposed architecture

This section presents our anonymous NDN-based FL architecture. We first define the proper naming scheme to achieve user anonymity in Section 5.1, then present the packet forwarding mechanism in Section 5.2. Section 5.3 presents how NDN caching is used in our approach, while Section 5.4 defines the enrollment mechanism that enables workers verification.

5.1. Naming scheme

For enabling the development of FL frameworks on top of NDN solutions, it is fundamental to define the set of names representing the interests to be satisfied. Indeed, naming conventions depend on the considered NDN application, e.g., vehicular networks [84]. In our context, the workers participating in the federation process need to publish their model and request the updated global model using unique identifiers. These identifiers depend on various aspects of the model to be published or requested, such as: *(i)* model location, i.e., local vs. global models; *(ii)* model versioning, i.e., which training round produced the model; and, *(iii)* workers anonymous identity.

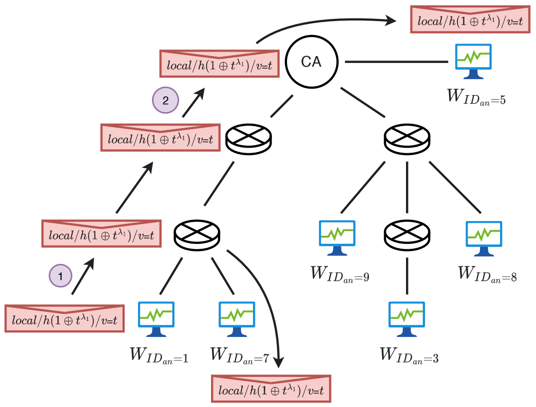
CA
$local/h(1 \oplus t^{\lambda_1})/v{=}t$
$local/h(1 \oplus t^{\lambda_1})/v{=}t$
2
$W_{ID_{an}=5}$
$local/h(1 \oplus t^{\lambda_1})/v{=}t$
$local/h(1 \oplus t^{\lambda_1})/v{=}t$
1
$local/h(1 \oplus t^{\lambda_1})/v{=}t$
$W_{ID_{an}=9}$
$W_{ID_{an}=8}$
$W_{ID_{an}=1}$
$W_{ID_{an}=7}$
$W_{ID_{an}=3}$
$local/h(1 \oplus t^{\lambda_1})/v{=}t$

Fig. 2. Name announcement procedure. (1) W_1 broadcasts a packet containing the anonymous name of its model. (2) CA receives W_1's announcement.

To obtain flexible names, we define a hybrid naming approach, where names are composed of different fields, each addressing one of the required information, i.e., location, versioning, and identity. On the other hand, to avoid anonymity leakage and satisfy R1, we cannot rely on plain-text naming schemes. Indeed, using an unprotected naming schema, e.g., *local/ID$_{an}$/v=t*, allows malicious entities to gather the information concerning the anonymous ID (ID_{an}) of the worker. In such a setup, malicious workers can pretend to be worker ID_{an} and advertise a different model addressed by $local/ID_{an}/v{=}t$. To protect against such scenarios, we leverage one-way hashing functions and a temporary secret factor to mask workers' anonymous identities. An example of the use of such names is proposed in Figs. 2–7. Here, the local model of worker having anonymous ID ID_{an} at the end of the 10th federation round can be addressed using the name: $local/h(ID_{an} \oplus s^{t=10}_{ID_{an}})/v{=}10$. The name contains the result of the hashing function h applied to the combination of the worker anonymous identity and a secret factor $s^{t=10}_{ID_{an}}$. The secret factor represents a temporary challenge known only by the worker ID_{an} and the CA. $s^{t=10}_{ID_{an}}$ can be arbitrarily complex depending on the security factor that must be achieved. As an example, consider the simple exponentiation operation shown in Eq. (2). Here, $s^{t=10}_{ID_{an}}$ depends on the local model timestamp $t = 10$ and on a secret parameter $\lambda_{ID_{an}}$ shared between worker ID_{an} and the CA during the worker enrollment procedure explained in Section 5.4.

$$h(ID_{an} \oplus s^{t}_{ID_{an}}) = h(ID_{an} \oplus t^{\lambda_{ID_{an}}}) \tag{2}$$

Thus, only the CA and worker ID_{an} can verify the overall hashing, given the challenge updates for each time the local model is published by worker ID_{an}. The combination of h and $s^{t}_{ID_{an}}$ allows us to protect the framework from impersonation attacks.

At the end of the 10th federation round, the aggregated model can be requested using the *global/v*=10 name. Here, the naming scheme does not require introducing the ID of the CA agent, as the global model can be advertised solely by the CA. Public-key cryptography can easily verify the global model's integrity and origin authentication. Here, the CA signs the global model using its private key k^{priv}_{CA}, while each worker verifies its content using k^{pub}_{CA}, as shown in details in Section 5.2. In such a way, our mechanism protects the federation against malicious workers aiming at impersonating the CA.

Interestingly, the proposed naming schema represents a flexible solution for multiple FL scenarios. Indeed, it allows the CA and workers to request any of the past local and global models, respectively. Such a requirement is fundamental in many relevant FL frameworks relying on model history for training optimisation, such as [85]. More complex naming schemes can also be identified depending on the underlying structure of the federation framework. For example, the need to introduce hierarchical naming approaches in the FL setups where workers can have a direct dependency [86] might exist. The required model can be addressed in this setup using an additional cluster field information inside the naming schema. Thus, the cluster field represents an optional field of the name, which can be extended depending on the hierarchical structure of the underlying federation. We might, for example, consider the local model of worker i belonging to cluster 3, itself belonging to cluster 1, to be represented using the clustered name *cluster1/cluster3/local/*$h(ID_{an} \oplus s^{t}_{ID_{an}})$*/v=t*. Meanwhile, the aggregated model obtained from the CA can be addressed using the general name *global/v=t* and the clusters' models can be referred to using the *global/cluster1/cluster3/v=t*

> **Synopsis.** Our naming schema follows the most used naming convention in vanilla NDN, i.e., hierarchical naming while anonymising the users' identities. However, the naming conventions are application-specific, allowing extensions and re-definitions.

5.2. Packet forwarding

Enabling FL in NDN is challenging as it needs to define a dedicated novel packet forwarding strategy. In particular, the most relevant issue is related to the pull-based nature of NDN networks where content requests are initialised by users looking for specific content. While suitable for user-centric Internet, NDN's nature represents a relevant drawback for push-based frameworks such as FL where clients and CA must periodically force their local model and global model to their counterparts.

To define an FL compliant NDN packet forwarding protocol, we require our novel forwarding mechanism to fulfil the following communication components:

1. Downward communication from the CA to worker, to which we refer as $CA \leftarrow W_{ID_{an}}$.
2. Downward communication from each worker to the CA, to which we refer as $W_{ID_{an}} \leftarrow CA$.
3. Upward communication from each client to the CA, to which we refer as $W_{ID_{an}} \rightarrow CA$.
4. Upward communication from the CA to each client, to which we refer as $CA \rightarrow W_{ID_{an}}$.

Next, we describe each of these components in detail, defining the NDN-based packet forwarding mechanism that enables them.

5.2.1. Downward communications ($W_{ID_{an}} \leftarrow CA$ and $CA \leftarrow W_{ID_{an}}$)

The communication respects the pull-based paradigm characterising NDN. Indeed, each downward communication requires the interested entity to download the required data through the communication medium. Therefore, in such a scenario, relying on the traditional NDN communication scheme is possible, where the interested party sends an interest request for a particular content. Here, the considered parties might be the CA or any worker belonging to the federation. Meanwhile, the requested content might be either the local model obtained via local training by a particular worker, i.e., $m^{(t)}_{ID_{an}}$ at $W_{ID_{an}}$, or the global aggregated model obtained by the CA, i.e., $M^{(t)}$. In the former, the CA issues an interest request packet for the name *local/*$h(ID_{an} \oplus s^{t}_{ID_{an}})$*/v=t*, following the naming schema presented in Section 5.1. Upon the reception of this interest packet, $W_{ID_{an}}$ replies to the CA's request forwarding $m^{(t)}_{ID_{an}}$, using a symmetric encryption scheme that is trusted between CA and $W_{ID_{an}}$. The model encryption process enables the verification of packet source, i.e., CA makes sure that $m^{(t)}_{ID_{an}}$ comes from $W_{ID_{an}}$ and not $W_{ID'_{an}}$, satisfying R2. Meanwhile, for the $W_{ID_{an}} \leftarrow CA$ case, the local worker interested in $M^{(t)}$ issues an interest request packet for the name *global/v=t*. Upon the reception of this interest, the CA signs the global model using the public-key cryptography scheme, allowing the verification of the model integrity and source authenticity (R2). Additionally, during the forwarding towards $W_{ID_{an}}$, the routers can cache the shared model for later use. The caching aspect is discussed more in detail in Section 5.3.

5.2.2. Upward communication from each worker to the CA ($W_{ID_{an}} \rightarrow CA$)

In FL, upon the completion of the local training procedure, each worker pushes its $m^{(t)}_{ID_{an}}$ to the CA for the aggregation process to execute. Vanilla NDN protocols to support this procedure have been recently proposed [27,28]. However, these protocols fail to meet R1, thus introducing the need for a novel push-enabling NDN packet forwarding protocol. Therefore, we define our ad-hoc mechanism enabling local workers to notify the CA and push their local models.

The overall mechanism is presented in Figs. 2–4, and is made of three components, namely: *(i) name announcement*, where each $W_{ID_{an}}$ announces the availability of its local update; *(ii) interest request*, where the CA issues the request for $m^{(t)}_{ID_{an}}$; and *(iii) update dispatching*, where $m^{(t)}_{ID_{an}}$ is forwarded to CA.

Name announcement. The announcement mechanism is implemented using a simple broadcasting scheme where $W_{ID_{an}}$ broadcasts a packet containing the name of the updated model: *local/*$h(ID_{an} \oplus s^{t}_{ID_{an}})$*/v=t*. Since the broadcasted information is received by all entities belonging to the federation network, and thus also other possibly malicious workers, the hashing applied to the ID_{an} is necessary to protect the network against impersonation attacks (R2). Moreover, such hashing is also helpful to allow the CA to verify the validity of the sponsored name. Indeed, upon its reception, the CA is capable of verifying the hashing $h(ID_{an} \oplus s^{t}_{ID_{an}})$ checking that it comes from a valid worker $W_{ID_{an}}$. The proposed announcement mechanism also presents a hidden advantage, as it allows the NDN routers that compose the underlying network to populate their FIB with the new available local model name. Populating the FIB allows for the routers to keep track of the path from $W_{ID_{an}}$ to CA so that consequent communications between such entities will follow the quickest path. Furthermore, FIBs can be implemented efficiently with reduced storage requirements [87], enabling the proposed strategy to scale even with many workers. A visual example of the proposed announcement mechanism can be found in Fig. 2.

Interest request. Upon the reception of the announcement message from $W_{ID_{an}}$ the CA initialises an interest request packet for the corresponding model name. To ensure the origin of the request packet, we require the CA to append to the interest packet a specific field containing a unique identity verification token, aiming at satisfying R2. Indeed, without an identity verification token, any other worker belonging to the federation would be allowed to reply to the announcement phase and issue an interest request for $m^{(t)}_{ID_{an}}$. We build the CA identity verification token τ_{CA} similarly to the hashing mechanism used for masking the worker's ID in the naming scheme (see Section 5.1). More in detail, we define $\tau_{CA} = h(k_{ID_{an}} \oplus \lambda_{ID_{an}})$, where $k_{ID_{an}}$ represents the shared symmetric key between the CA and $W_{ID_{an}}$, obtained during enrollment (see Section 5.4). Therefore, the CA issues the request as the packet $I(local/\mathrm{h}(\mathrm{ID}_{\mathrm{an}} \oplus \mathrm{s}_{\mathrm{ID}_{\mathrm{an}}}{}^{\mathrm{t}})/\mathrm{v}=\mathrm{t}) \| h(k_{ID_{an}} \oplus \lambda_{ID_{an}})$, where $I(name)$ represents the standard interest packet for certain content, and $\|$ represents the concatenation operation. The interest request mechanism is detailed in Fig. 3.

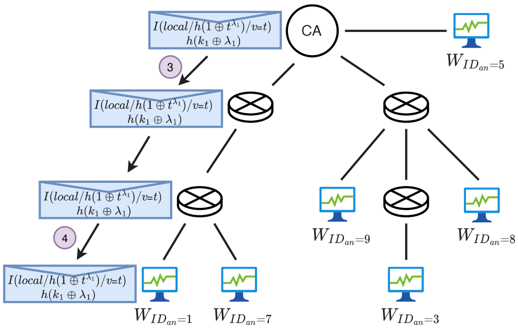

Fig. 3. Interest request procedure. ③ CA sends interest request towards W_1. ④ The interest request packet follows the stored inverse path to reach back W_1.

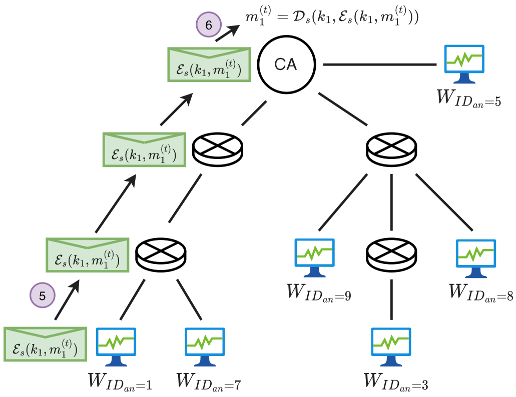
$m_1^{(t)} = \mathcal{D}_s(k_1, \mathcal{E}_s(k_1, m_1^{(t)}))$
6
5
$\mathcal{E}_s(k_1, m_1^{(t)})$
CA
$W_{ID_{an}=5}$
$W_{ID_{an}=9}$
$W_{ID_{an}=8}$
$W_{ID_{an}=1}$
$W_{ID_{an}=7}$
$W_{ID_{an}=3}$

Fig. 4. Update dispatching procedure. ⑤ W_1 sends the encrypted local update $\mathcal{E}_s(k_1, m_1^{(t)})$ towards CA. ⑥ CA deciphers the local update received from W_1.

Update dispatching. $W_{ID_{an}}$ checks the identity of the CA validating the token τ_{CA}. If the interest request is valid, $W_{ID_{an}}$ forwards $m^{(t)}_{ID_{an}}$ as a reply to the interest request, following the standard NDN paradigm. To protect the dispatch of $m^{(t)}_{ID_{an}}$ from any eavesdropper (R2) and avoid caching mechanisms to cache local models, we require the local worker to encrypt the update using the symmetric key $k_{ID_{an}}$ shared between $W_{ID_{an}}$ and CA during enrollment (see Section 5.4). Mathematically, $W_{ID_{an}}$ forwards $\mathcal{E}_s(k_{ID_{an}}, m^{(t)}_{ID_{an}})$, where $\mathcal{E}_s$ represents any symmetric encryption scheme and $k_{ID_{an}}$ the corresponding key shared between $W_{ID_{an}}$ and the CA. Meanwhile, the CA can decipher the local update following $m^{(t)}_{ID_{an}} = \mathcal{D}_s(k_{ID_{an}}, \mathcal{E}_s(k_{ID_{an}}, m^{(t)}_{ID_{an}}))$, where $\mathcal{D}_s$ is the decryption scheme corresponding to $\mathcal{E}_s$. The symmetric encryption mechanism is fundamental for preventing routers from caching the local update, thus protecting NDN-based FL against any cache probing attacks. An example of update dispatching is presented in Fig. 4.

Following the proposed approach, the obtained $W_{ID_{an}} \rightarrow CA$ upward communication becomes secure and anonymous. Indeed, the hashing trick introduced in NDN names makes it impossible for malicious workers to gather information concerning the anonymous identity of other workers and impersonate them. Moreover, the use of the CA entity verification token τ_{CA} makes it impossible for workers to impersonate the CA and request a local update. The same τ_{CA} is also secure against attackers interested into obtaining $k_{ID_{an}}$ or $\lambda_{ID_{an}}$ from the CA interest request. Indeed, reversing the hashing $h(k_{ID_{an}} \oplus \lambda_{ID_{an}})$ is proven to be complex, and even if broken it would be complex for an attacker to get $k_{ID_{an}}$ or $\lambda_{ID_{an}}$ from $k_{ID_{an}} \oplus \lambda_{ID_{an}}$. Finally, the symmetric encryption avoids local model caching, which might be a threat due to the few attacks available against NDN caching paradigms [88,89].

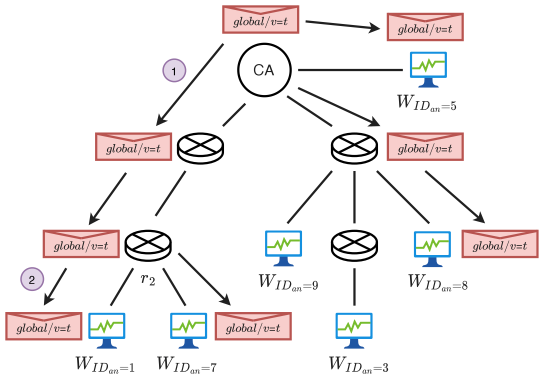

Fig. 5. CA's name announcement procedure. ① CA broadcasts a packet containing the name of the aggregated global update. ② W_1 and other workers receive CA's announcement.

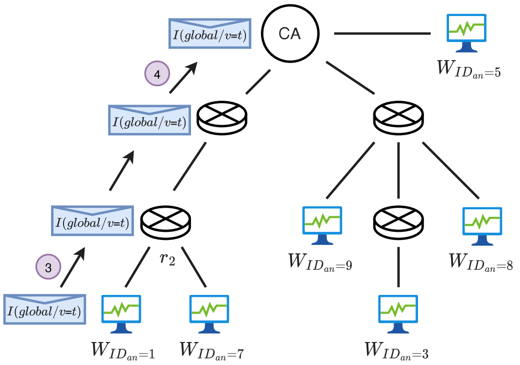

Fig. 6. $W_{ID_{an}}$'s interest request for $M^{(t)}$. ③ W_1 sends interest request towards CA. ④ The interest request packet follows the stored inverse path to reach back CA.

5.2.3. Upward communication from CA to each worker ($CA \rightarrow W_{ID_{an}}$)

Similarly to Section 5.2.2, we here design the upward communication scheme from the CA to each worker, enabling the CA to push $M^{(t)}$ to $W_{ID_{an}}$. An example for $CA \rightarrow W_{ID_{an}}$ is shown in Figs. 5–7.

Name announcement. CA announces $M^{(t)}$ availability, sending in broadcast a packet containing the name of the global model (*global/v=t*). This name announcement follows a plain-text mechanism that does not require the introduction of hashing to mask the origin identity of the broadcast, as shown in Fig. 5.

Interest request. Each local worker issues an interest request packet for $M^{(t)}$, which we express under the notation $I(global/v{=}t)$, as shown in Fig. 6. Here, defining any encryption criteria or identity verification mechanisms for the workers to follow while emitting their interest requests is irrelevant. Indeed, all the workers belonging to the federation are interested in requesting the updated global model; thus, all enrolled workers are authorised to access it.

Update dispatching. The interest requests issued by the different workers end up at the CA, which replies by dispatching the updated model. The packets containing the dispatched $M^{(t)}$ must now be encrypted using an asymmetric encryption scheme. Indeed, the workers need to establish the correct origin of $M^{(t)}$ to avoid CA impersonation attacks (R2). Moreover, only workers belonging to the federation, and thus owning the public key of CA, should be allowed to receive $M^{(t)}$. Indeed, the removal of this asymmetric encryption would allow any node outside the federation to request and obtain $M^{(t)}$. Finally, the use of asymmetric encryption enables the possibility for the routers of the NDN network to cache $M^{(t)}$, as it is possible to cache its encrypted version. Enabling caching of the global model allows its fast

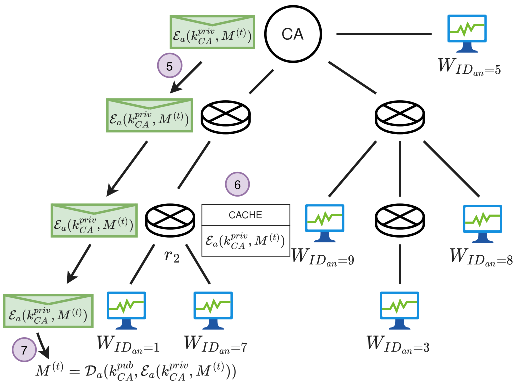
CA
$\mathcal{E}_a(k_{CA}^{priv}, M^{(t)})$
5
$W_{ID_{an}=5}$
$\mathcal{E}_a(k_{CA}^{priv}, M^{(t)})$
6
$\mathcal{E}_a(k_{CA}^{priv}, M^{(t)})$
CACHE
$\mathcal{E}_a(k_{CA}^{priv}, M^{(t)})$
r_2
$W_{ID_{an}=9}$
$W_{ID_{an}=8}$
$\mathcal{E}_a(k_{CA}^{priv}, M^{(t)})$
7
$W_{ID_{an}=1}$
$W_{ID_{an}=7}$
$W_{ID_{an}=3}$
$M^{(t)} = \mathcal{D}_a(k_{CA}^{pub}, \mathcal{E}_a(k_{CA}^{priv}, M^{(t)}))$

Fig. 7. $M^{(t)}$ dispatching. ⑤ CA sends the encrypted global update $\mathcal{E}_a(k_{CA}^{priv}, M^{(t)})$ towards W_1. ⑥ The encrypted global model is stored in routers caches for further workers requests. ⑦ W_1 deciphers the global update received from CA.

distribution amongst all the workers. Therefore, the proposed scheme is expected to reduce the latency when distributing the updated global model amongst the workers (see Section 7). Fig. 7 shows an example of update dispatching.

The proposed $CA \rightarrow W_{ID_{an}}$ forwarding mechanism satisfies R2 as it guarantees the origin of the updates concerning the global model, i.e., only CA can send $M^{(t)}$ correctly. Meanwhile, we ensure that only authorised workers can access the global model information, avoiding information leakage outside the federation. Moreover, the anonymity of the workers belonging to the federation is untouched, satisfying R1, as each $W_{ID_{an}}$ does not share any personal information, but rather issues only interest requests to the CA. Finally, the proposed packet forwarding scheme does not introduce any requirements on the shape of the local and global updates during their dispatch. Therefore, our approach is compatible with any *data privacy* mechanisms manipulating updated content, such as DP, thus satisfying R3.

> **Synopsis.** Our novel packet forwarding mechanism solves the demand for a push-based communication for FL in an NDN environment while satisfying the anonymity and security requirements R1 and R2. To this end, we introduce an anonymised name broadcasting procedure relying on a shared secret key and entity verification token.

5.3. Caching

Caching represents one of the most innovative features of NDN, introduced to reduce access latency for popular content. Nevertheless, the presence of caches has also been proven to be a possible point of attack for privacy disclosure [90,91]. Therefore, while designing the caching requirements of any NDN-based framework, it is relevant to predict and prevent possible leakage of information that might arise. Thus, while defining our caching protocol, we consider the following aspects:

A1 - Local updates in the FL framework must be kept anonymous and private.
A2 - The distribution of the global updates requires sharing the same information, i.e., $M^{(t)}$, among all workers.
A3 - Global updates must be accessible only to federation participants.

A1-A3 highlight different requirements of the underlying FL framework and give relevant hints concerning the caching protocol required. More in detail, A1 represents faithfully the anonymity concerns that affect any FL setup and highlights the necessity for avoiding caching of local models $m_{ID_{an}}^{(t)}$. Indeed, local updates should be shared solely between $W_{ID_{an}}$ and the CA. Moreover, NDN caches introduce a possible lack of information and might be subject to cache tampering attacks. In such a scenario, malicious workers can extract relevant information concerning the cached $m_{ID_{an}}^{(t)}$ local model by attacking the intermediate routers. Thus, to avoid this issue and satisfy R1 and R2, we here propose to not cache any of the local models sent from each worker to the CA. Here, it is also relevant to notice that NDN caching paradigm was first introduced to boost the communication performance of the network, aiming to cache popular content reducing its latency. In any standard federation framework, local models $m_{ID_{an}}^{(t)}$ must not represent popular content, as they are sent only once from each worker in each federation round. Therefore, introducing caching principles over the local updates does not improve latency and only hinders the $W_{ID_{an}}$'s anonymity.

While caching during the local model distribution does not represent a desirable approach, the same does not hold when dealing with the distribution of the global updates. Indeed, A2 stresses the possible achievable benefits of communication efficiency. At the end of each federation round, the global update from CA to $W_{ID_{an}}$ represents the most popular content in the federation. Caching such updates inside intermediate routers may drastically reduce the latency a worker requires to access the content. As an example, consider two workers placed in the same local network ($W_{ID_{an}=1}$ and $W_{ID_{an}=7}$ in Fig. 7). One of these two workers receives a response for its interest packet directly from its border router (r_2 in Fig. 7), rather than communicating with the, perhaps very distant, CA.

Given these latency improvements, we design our caching protocol to allow the caching of global updates. A3 shows possible drawbacks that might arise from a naive caching of CA's updates. Indeed, information concerning the result of the aggregation process in FL should be accessible only to workers belonging to the federation. Recall from Section 5.2 that our mechanism employs an asymmetric encryption scheme while forwarding global updates from the CA to each worker. The global updates are encrypted using the CA private key before being distributed and cached. As CA's asymmetric key is distributed to workers during the enrollment phase (see Section 5.4), only authorised workers belonging to the federation can decipher cached updates, thus satisfying R2. Mathematically, we require the CA and the intermediate routers to send and cache $\mathcal{E}_a(k_{CA}^{priv}, M^{(t)})$, where $\mathcal{E}_a$ represents any asymmetric encryption function and k_{CA}^{priv} the corresponding CA's private key. These updates are deciphered by each worker using $M^{(t)} = \mathcal{D}_a(k_{CA}^{pub}, \mathcal{E}_a(k_{CA}^{priv}, M^{(t)}))$, where $\mathcal{D}_a$ represents the decryption process of the asymmetric encryption function $\mathcal{E}_a$ and k_{CA}^{pub} the corresponding CA's public key.

This process is shown in Fig. 7. The proposed caching protocol satisfies A1-A3, enabling latency benefits while avoiding information leakage or anonymity infringement.

> **Synopsis.** Enabling caching in FL frameworks is beneficial for communication efficiency, especially in large federation setups. Here, vanilla NDN facilitates caching of global updates while requiring negligible effort.

5.4. Enrollment scheme

In this section, we design a lightweight mechanism for the enrollment and authentication of a new worker who aspires to participate in the federation. To satisfy R1, for each worker with real identity, say ID_i, that requests to join the federation, the TTP assigns an anonymous identity, say ID_{an}. Instead, to meet R2, we use Physically Unclonable Functions (PUFs) in the worker enrollment phase. PUFs are mainly used as a hardware-specific security element for the device's cryptography. They are created during the manufacturing phase of each device, making them impossible to clone or replicate. In this context, PUFs are considered the physical identity of the electronic devices, making them widely used in cryptographic tasks. In particular, PUFs performs some functional functions, i.e., for each query for a given input, it produces an output. Given the PUFs function $\mathcal{F}_{puf}$ and the input as a *challenge (chal)*, the output as *response (O)* can be calculated as: $O = \mathcal{F}_{puf}(chal)$ Generally, for the same challenge, two PUFs must always output different responses while always responding equivalently to the same challenge. However, this is only sometimes the case, as the response generation algorithm is sensitive to noisy environments. Encountering such an issue, the response generation and restore algorithms are modified, encompassing a fuzzy extractor and an Error Correcting Code (ECC) [92]. The response generation algorithm, depicted in Algorithm 1, is used in the worker both in the setup and enrollment phase. Here, the algorithm outputs the response value to be recovered, i.e., r, and a public helper string, i.e., P, which is used to retrieve the response. The ECC corrects up to κ errors in subsequent PUFs outputs to deal with noisy environments.

Algorithm 1 Response Generation Algorithm

```
Input: Modulus $n$, Challenge $chal$
Output: $\langle r, P \rangle$
  $O = \mathcal{F}_{puf}(chal)$;
  $r \xleftarrow{Sample} \mathbb{Z}_n$;
  $P = O_r \oplus ECC(r)$
  return $\langle r, P \rangle$
```

Instead, the response restore algorithm, depicted in Algorithm 2, allows an output generated by PUFs if it differs from the original output by at most κ bits. The procedure restores the response r using the public helper string P and the error decoding algorithm D.

Algorithm 2 Response Restore Algorithm

```
Input: Challenge $chal$, Public Helper String $P$
Output: $r$
  $O' = \mathcal{F}_{puf}(chal)$;
  $r = D(P \oplus O'_r)$
  return $r$
```

Using PUFs, we can prevent the malicious workers that aim at impersonating another worker after having eavesdropped its ID_{an}. Furthermore, the use of PUFs supports the CA for preventing the malicious workers that aim at joining multiple times with different identities. Lastly, our scheme of enrollment and authentication satisfies R2. In particular, avoiding using the worker's real identity and anonymous identity in the communication prevents the honest but curious CA from correlating the models to a specific worker.

5.4.1. Setup phase

In the setup phase, our mechanism relies on a trusted third party, say TTP, that verifies a worker W_i that wishes to join the federation. In particular, the TTP is assumed to be a trusted entity that knows the PUFs, similarly to [93–95]. Being a trusted entity ensures the reliability of the verification for each new node while maintaining its identity anonymous. A synopsis of the setup phase is presented in Fig. 8 and summarised in the following steps:

1. W_i chooses its identity related information, i.e., ID_i, and shares it with the TTP.
2. TTP initialises a challenge-response phase. Here, for each challenge, i.e., $chal_k \in ChalSet$, received by the TTP, W_i generates a response, i.e., res_k, and propagates it to the TTP which stores each challenge and the respective response in a table, i.e., $tab\langle chal, res\rangle$.

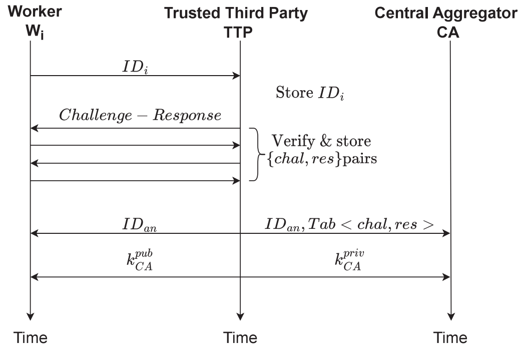

Fig. 8. Setup phase.

3. Once TTP verifies the W_i, it generates an anonymous identity, i.e., ID_{an}, that is used for the future communication. Thereafter, TTP disseminates ID_{an} to W_i and $ID_{an}, tab\langle chal, res\rangle$ to the CA.
4. Lastly, the TTP generates the asymmetric keys k^{priv}_{CA} and k^{pub}_{CA} for the CA used for the encryption and decryption process. After that, the TTP disseminates k^{pub}_{CA} and k^{priv}_{CA} to the worker W_i and CA, respectively.

For the challenge-response phase, we consider W_i uses Algorithm 1 to generate the response on the challenge based on its PUFs. After that, the TTP adopts Algorithm 2 to verify the correctness of W_i's response. Once the TTP validates the responses, it maps the challenge and response pairs in a table and securely stores it for checking misbehaviour in future enrollments.

5.4.2. Enrollment phase

In this phase, the worker W_i, previously verified by the TTP, wishes to be anonymously enrolled to the federation by the CA. In our mechanism, the enrollment phase depicted in Fig. 9, includes the steps described subsequently:

1. W_i submits a registration request to the CA using its anonymised identity ID_{an}.
2. CA picks a random challenge $chal_j \in ChalSet$ and propagates it to W_i.
3. W_i generates a response res_j for the challenge and propagates it to the CA.
4. CA verifies whether the res_j hits one of the entries of the table provided by TTP. If this is the case, W_i is enrolled. Otherwise, its request to join the federation is rejected.
5. CA generates a symmetric key $k_{ID_{an}}$ to be used for the communication with W_i and the secret parameter $\lambda_{ID_{an}}$ that is used to calculate the name hashes as in Eq. (2).

Similarly to the setup phase, here W_i uses Algorithm 1 to generate the response. Once the CA and W_i share the symmetric key and the secret parameter, the enrollment phase is concluded.

Synopsis. Our enrollment mechanism requires the use of TTP and PUFs to correlate a worker to its anonymous identity uniquely. These two pillars enable an anonymous join in the federation while preventing malicious behaviours.

6. Anonymity discussion

In this section, we provide an extensive discussion on the robustness of our mechanism against threats identified in Section 4.2. Here, we elaborate on the malicious workers' threat, including worker impersonation and, in and out of federation eavesdropping in Section 6.1. Then, we consider the threat from CA posing rather as an honest but curious or malicious entity in Section 6.2.

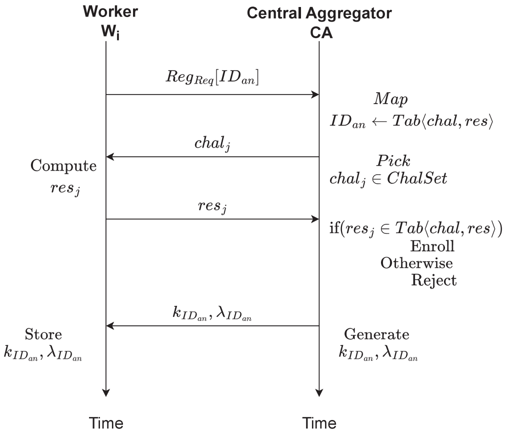
Worker
W_i
Central Aggregator
CA
$Reg_{Req}[ID_{an}]$
Map
$ID_{an} \leftarrow Tab\langle chal, res\rangle$
$chal_j$
$Pick$
$chal_j \in ChalSet$
Compute
res_j
res_j
if($res_j \in Tab\langle chal, res\rangle$)
Enroll
Otherwise
Reject
$k_{ID_{an}}, \lambda_{ID_{an}}$
Store
$k_{ID_{an}}, \lambda_{ID_{an}}$
Generate
$k_{ID_{an}}, \lambda_{ID_{an}}$
Time
Time

Fig. 9. Enrollment phase.

6.1. Malicious workers

In federation eavesdropping. Hindering the possibility for workers to eavesdrop on local updates, our framework relies on multiple components. We first define name announcements – corresponding to local updates – to be anonymous and verifiable only by the CA. Such names, however, could be requested by any node upon the reception of the name announcement packet. Therefore, to avoid malicious clients requesting and obtaining local updates of other workers, we require the local update requests to contain an identifier field. We define our identity verifier field as $h(k_{ID_{an}} \oplus \lambda_{ID_{an}})$. Since it is built over $k_{ID_{an}}$ and $\lambda_{ID_{an}}$, such a field can be adequately constructed only by the worker and the CA. Therefore, malicious workers can only eavesdrop on other clients' updates only if they know their $k_{ID_{an}}$ and $\lambda_{ID_{an}}$. Finally, the local updates are sent by each worker using a symmetric encryption scheme with $k_{ID_{an}}$ as the key. Thus, malicious clients cannot extract information from the exchanged packets nor tamper the routers' caches, as the local models are not cached.

Out of federation eavesdropping. The proposed NDN based protocol relies on caching global updates to optimise the communication between CA and each worker. However, this caching introduces issues concerning the security of the global models being accessible from outside the federation via cache probing mechanisms [88]. To fix such an issue, we introduce the need for an asymmetric encryption scheme between the CA and the federation clients whenever the global updates are shared. Indeed, our mechanism requires sharing and caching the encrypted version of the local updates. Therefore, we protect the framework from out-of-federation eavesdropping nodes.

Worker impersonation. FL frameworks that avoid considering workers' anonymity are susceptible to impersonation attacks. In this scenario, malicious workers belonging to the federation can eavesdrop on other workers' identities, e.g., IP addresses or similar, and pretend to be them spoofing their identity. In such a way, an attacker can send corrupted updates, linking them with an honest user. Here, the key to blocking such malicious behaviour relies on making the federation clients incapable of eavesdropping on other workers' identities, and anonymising them. Additionally, it is fundamental to notice that even the anonymous identity must be kept impossible to eavesdrop. Indeed, in NDN, eavesdropping on the anonymous identity via content name taping would allow malicious clients to announce malicious updates on behalf of the eavesdropped identity. Therefore, to tackle such an issue, we introduced a novel naming schema to hide the anonymous identity of the federation's workers. Indeed, at each federation round, the announced names rely on hashing a secret shared only between the CA and the worker announcing the name corresponding to the update. Thus, the names can be verified only by the CA and $W_{ID_{an}}$ involved in the communication. A malicious client would be able only to eavesdrop on the hashed identity of the worker $h(ID_{an} \oplus s^t_{ID_{an}})$. Therefore, to extract ID_{an} from the hashing result, an attacker must invert h and $\oplus$. The attack success probability can be written as:

$$Pr(success) = Pr(h^{-1}(ID_{an} \oplus s^t_{ID_{an}})) \cdot Pr(ID_{an} | ID_{an} \oplus s^t_{ID_{an}}) \tag{3}$$

where $Pr(h^{-1}(ID_{an} \oplus s^t_{ID_{an}}))$ represents the probability of being successful in inverting the hashing function, while $Pr(ID_{an} | ID_{an} \oplus s^t_{ID_{an}})$ represents the probability of being capable of extracting ID_{an} from the $\oplus$ result. The attack success probability can be tailored to be smaller than a given security factor ϵ by selecting a proper h. Deriving from Eq. (3), we require h such that:

$$Pr(h^{-1}(ID_{an} \oplus s^t_{ID_{an}})) \leq \frac{\epsilon}{Pr(ID_{an} | ID_{an} \oplus s^t_{ID_{an}})}. \tag{4}$$

6.2. Honest but curious or malicious CA

In our framework, we achieve unlinkability between forwarded model updates in FL and their corresponding source identity. To this end, the proposed system ensures that the CA does not have knowledge concerning the real identity of the federation clients but only knows their anonymous IDs provided by the TTP. Besides, given the location-independent nature of our NDN-based FL framework, the CA cannot infer where the worker is located in the network. Thus, even considering a CA agent capable of reconstructing the local data from the received updates, we ensure that the CA cannot link such data to the worker's real identity. Moreover, it is essential to notice that the proposed anonymity mechanism does not impose any requirement on the shape of the local updates and thus can be coupled with any DP technique. Therefore, the combination of our framework with DP allows for avoiding data disclosure, rendering it cumbersome to both reconstruct data from updates (data privacy) and link such data to a specific identity (user anonymity). Since the CA does not know the real identity of the federation clients, the CA cannot extract targeted local data information, i.e., honest but curious CA, nor to corrupt target updates, i.e., malicious CA.

7. Performance evaluation

In this section, we evaluate the proposed NDN-based protocol, comparing it against a clean-slate IP-based protocol and many state-of-the-art anonymity solutions. We emphasise that since our novel protocol does not impact the model optimisation process, our experiments focus solely on the communication aspects of the federation. Also, model performance metrics, e.g., accuracy, F1 score, are not impacted by our solution and, hence, are not evaluated. Therefore, we evaluate the proposed communication protocol in terms of the communication latency incurred to perform a certain number of FL aggregations.

7.1. Simulation setup

We simulate multiple networks where each worker node W_i has a certain distance from the CA in terms of number of hops, n_i^{hops}. At the beginning of each FL run, we draw the number of hops for each worker to be a uniform random number between a minimum of one and a maximum of ten. To each hop, we assign a random bit rate B^{hop} $[\frac{\text{bits}}{\text{s}}]$ drawn from a range of possible rates. We also simulate a hop latency T^{hop} in addition to the load transmission delay via an exponential random variable. The resulting delay D^{hop} associated to a given hop can be written as

$$D^{\text{hop}} = \frac{P_S}{B^{\text{hop}}} + T^{\text{hop}} \quad [\text{s}], \tag{5}$$

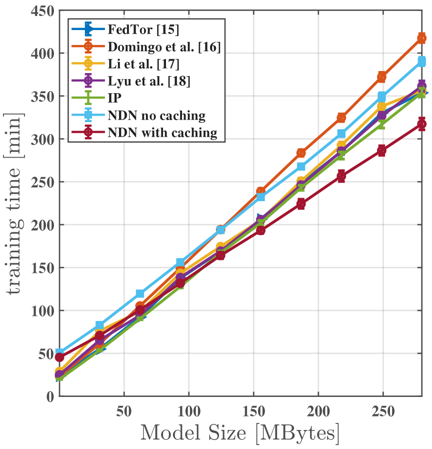

Fig. 10. Performance of the proposed algorithm regarding the total training time to perform 30 aggregation steps. We show the training time versus the model size for $N = 200$.

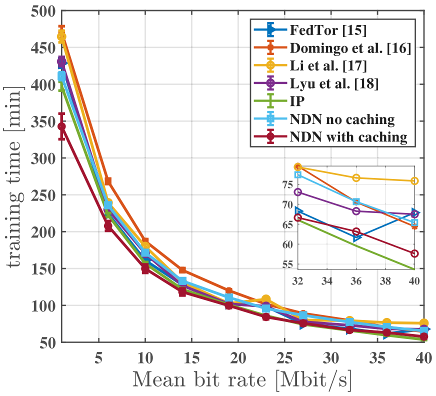

Fig. 12. Performance of the proposed algorithm regarding the total training time to perform 30 aggregation steps. We show the time versus the mean bit rate of the hops. The number of workers is fixed to $N = 50$, and the model size is $S = 200$ [MBytes].

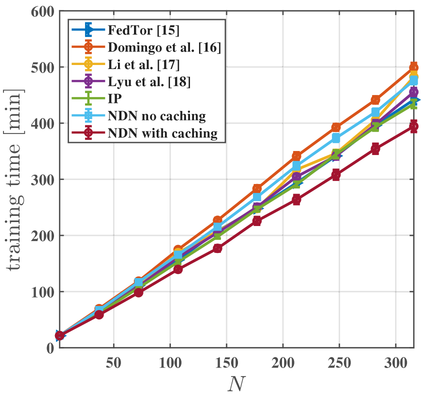

Fig. 11. Performance of the proposed algorithm regarding the total training time to perform 30 aggregation steps. We fix the model size $S = 200$ [MBytes] and show the training time versus the number of participating devices, N.

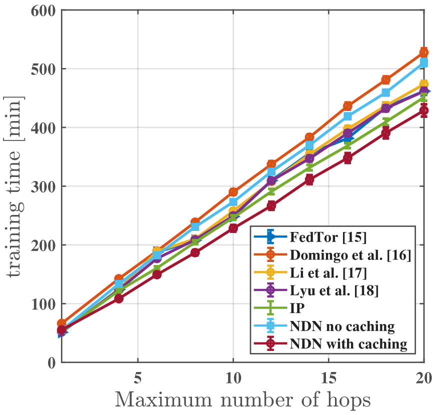

Fig. 13. Performance of the proposed algorithm regarding the total training time to perform 30 aggregation steps. We fix the model size $S = 200$ [MBytes] and the number of workers $N = 200$. We show the training time versus the maximum number of hops between workers and CA.

where P_S is the packet size in bits. We consider the same packet size of 65 KBytes for IP and NDN packets. We set the header to be 40 Bytes for IP, while we set it to 72 Bytes for NDN to include the signature and hashing overheads. Indeed, the basic NDN header is 8 Bytes, to which we add 64 Bytes to account for the content name and the hashing function.

Simulating network delays via exponential random variables is a well-established network modelling technique [96]. To model the increase in latency when the number of participating nodes N increases, we modify the expectation of the exponential random variables as $T^{\text{hop}} \sim \exp(c(N)) \quad [\text{s}]$, where $c(N)$ is a function of the number of participating nodes. A reasonable choice is to pick $c(N)$ to be monotonically increasing with N. Indeed, if the N increases, the network congestion is expected to increase. For example, in a 5G network, $c(N)$ could be influenced by the virtual network resources associated with the FL service. For simplicity, we perform our simulations with $c(N) = kN$, with k constant. This models a hop latency whose expectation grows linearly with the number of participating nodes. Similarly, we model the per-hop bit rates as a decreasing function of N. Given a reference bit rate range $[B_1, B_2]$, we let

$$B^{\text{hop}} \sim \frac{1}{N}\mathcal{U}[B_1, B_2] \;\; [\text{Mbit/s}]. \tag{6}$$

To simulate caching, we associate a probability of cache hit to each worker at each aggregation round, for the $CA \rightarrow W_i$ upward communication from the aggregator to the workers. The caching probability is modelled as an increasing function of the number of participating workers. Such a choice is motivated by the fact that when more network nodes are involved in the federation, a cache hit in some intermediate hop is more likely to occur. Given the number of participating workers N, we write the caching probability as $\mathbb{P}[\text{caching} \mid \# \text{ nodes} = N] = 1 - g(N)$, where $g(N) \in [0, 1]$ is a monotonically non-increasing function of N. Examples of $g(N)$ we experimented with are $g(N) = 1/\sqrt{N}$ or $g(N) = 1/N$. A hit in the cache reduces the number of hops between the CA and a worker. We simulate this reduction in the number of hops for a worker W_i by randomly drawing a positive integer in $\{1, \dots, n_i^{\text{hops}}\}$, where n_i^{hops} is the original number of hops between the CA and W_i.

7.2. Simulation results

This section compares our solution with a basic IP configuration and several state-of-the-art anonymous solutions. For each communication protocol baseline, we model all the additional communication components producing overhead that characterise such a solution, including notification messages and encryption processes. More in detail, for Domingo et al. [16], we include the time required to transmit the model to another worker, as required by the protocol. Each model packet is forwarded to a different worker before being delivered to the CA, thus introducing high overhead. Meanwhile, methods relying on an onion-style encryption scheme to guarantee anonymity [15,17,18] require additional encryption mechanisms at each node. Therefore, we model their additional overhead, adding an encryption latency for each node involved in the communication path, and model it as a random exponential delay. Finally, we also study two versions of the proposed NDN-based protocol, implementing our solution with and without caching. Here, we include all the intermediate communication rounds required to enable the proposed push-based mechanism, while caching is modelled as hop savings in the communication. The simulation results are obtained using $g(N) = 1/\sqrt{N}$ and $k = 0.05$. We denote the model size by S.

The results shown in Figs. 10, 11 and 13 have been obtained simulating transmission bit rates randomly varying between a minimum of 1 Mbit/s and a maximum of 100 Mbit/s. Throughout our experimental evaluation, we show the average performance obtained over multiple experiments and the corresponding 0.95 confidence intervals. We first study the impact of the ML model size on the communication performance, letting the model size S vary between 1 and 250 MBs while the number of workers is fixed at $N = 200$. The results in Fig. 10 highlight the increased benefits obtained from NDN caching when the model grows. The proposed protocol reaches performance improvements of up to 30% w.r.t. baselines.

Fig. 11 presents the results of analysing the communication performance against various workers N. Here, we keep the model size $S = 200$ MBs. The results highlight the superiority of our solution for all setups. Moreover, a linear increase of improvements exists as the number of participating workers N grows, reaching a ∼20% training time reduction for $N = 200$. The proportionality between N and training time improvements is due to the increase in the probability of cache hit. Thanks to the increased number of participating nodes, a higher number of cache hits occurs, reducing the communication latency between CA and workers. We also analyse the impact of bit rates between hops on communication effectiveness. Here, we vary the average bit rate of hops between 1 and 40 Mbit/s while keeping $S = 200$ MBs and $N = 200$. Fig. 12 shows the results of our analysis. The proposed NDN-based protocol outperforms the state-of-the-art approaches while achieving comparable performance with vanilla-IP. The obtained performance improvements are particularly evident for low bit rate values, reaching up to ∼25% lower training time. Finally, we analyse the impact of the communication path length on the achieved training time, varying the maximum number of hops from each worker to the CA between 1 and 20. Here, we set $S = 200$ MBs and $N = 200$. Fig. 13 provides the results. As expected, the proposed solution outperforms state-of-the-art overall setups thanks to its caching capabilities. Similarly to Fig. 11, the performance improvement increases with more hops in the communication path, reaching improvements of up to ∼20%. The obtained results highlight the superiority of NDN-based solution against the state-of-the-art frameworks over all the setups. Interestingly, these performance improvements are obtained despite NDN requiring a larger packet header w.r.t. IP-based solutions. Indeed, NDN header includes the required bytes to represent the content name and their hashed values. Therefore, the packet header size does not significantly impact the training time performance in FL setups.

8. Conclusions and future works

In this paper, we tackle the lack of anonymity issue of the FL frameworks, which threatens overall federation privacy. We propose a novel NDN-based federation scheme to overcome this issue, enabling anonymous-by-design communication in FL. To this aim, we extend the vanilla-NDN protocol to be FL compliant and fully anonymous by adding: *(i)* a new-fashioned naming convention; *(ii)* a novel packet forwarding protocol that encompasses both pull and push-based models that FL seeks for; and *(iii)* a lightweight enrollment scheme. The novel framework is designed to satisfy several anonymity and security requirements common in FL setups, such as identity anonymity and robustness to misbehaving parties. To evaluate how and to what extent our mechanism satisfies such requirements, we thoroughly discuss its robustness against well-known FL security and privacy issues. The performance of our novel framework is analysed from a communication efficiency perspective, comparing our mechanism to multiple baselines. The results show improved performance in terms of overall training time with respect to various parameters of the federation mechanism against all selected baselines. The findings of this paper highlight the relevance of ensuring anonymity-by-design in FL scenarios where multiple competitor parties are involved in the federation. The proposed framework shows the benefits of leveraging a communication protocol designed with anonymity and privacy in mind, such as NDN.

Future works include the implementation and deployment of the proposed mechanism over an NDN testbed, aiming at showing its advantages compared to IP-based anonymity frameworks over a broad range of ML and DL scenarios such as computer vision [97–99], graph processing [100–102], and neuro-symbolic integration [103,104]. The real-world deployment of the proposed scheme allows the FL and NDN research communities to rely on anonymity-by-design communication and extend the proposed framework, ensuring safely distributed optimisation of ML models. Finally, we intend to integrate our anonymity mechanism with available data privacy solutions to present the first completely private FL framework.

CRediT authorship contribution statement

Andrea Agiollo: Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing. **Enkeleda Bardhi:** Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing. **Mauro Conti:** Visualization, Supervision. **Nicolò Dal Fabbro:** Conceptualization, Software, Investigation, Writing – original draft. **Riccardo Lazzeretti:** Visualization, Supervision.

Declaration of competing interest

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

Data availability

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

Acknowledgements

This paper was partially supported by *(i)* La Sapienza University of Rome within the Bando Ricerca 2020, project "Secure ANd PrivatE Information sharing (SANPEI)"; *(ii)* project "SERICS" (PE00000014) under the NRRP MUR program funded by the EU - NGEU. We would also like to thank the anonymous reviewers for their comments and suggestions that helped us improve the quality of the paper.

References

[1] J. Konečný, H.B. McMahan, F.X. Yu, P. Richtárik, A.T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, CoRR, abs/1610.05492, 2016.

[2] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, H. Qi, Beyond inferring class representatives: User-level privacy leakage from federated learning, in: IEEE Conference on Computer Communications, INFOCOM 2019, Paris, France, April 29 - May 2, 2019, IEEE, 2019, pp. 2512–2520, http://dx.doi.org/10.1109/INFOCOM.2019.8737416.

[3] L. Zhu, Z. Liu, S. Han, Deep leakage from gradients, in: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December (2019) 8-14, Vancouver, BC, Canada, 2019, pp. 14747–14756, https://proceedings.neurips.cc/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html.

[4] B. Hitaj, G. Ateniese, F. Pérez-Cruz, Deep models under the GAN: information leakage from collaborative deep learning, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, ACM, 2017, pp. 603–618, http://dx.doi.org/10.1145/3133956.3134012.

[5] J. Geiping, H. Bauermeister, H. Dröge, M. Moeller, Inverting gradients - how easy is it to break privacy in federated learning? in: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December (2020) 6-12, Virtual, 2020, https://proceedings.neurips.cc/paper/2020/hash/c4ede56bbd98819ae6112b20ac6bf145-Abstract.html.

[6] J. Zhang, B. Chen, X. Cheng, H.T.T. Binh, S. Yu, PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems, IEEE Internet Things J. 8 (2021) 3310–3322, http://dx.doi.org/10.1109/JIOT.2020.3023126.

[7] Y. Ishai, E. Kushilevitz, R. Ostrovsky, A. Sahai, Cryptography from anonymity, in: 47th Annual IEEE Symposium on Foundations of Computer Science FOCS 2006, 21-24 2006, Berkeley, California, USA, Proceedings, IEEE Computer Society, 2006, pp. 239–248, http://dx.doi.org/10.1109/FOCS.2006.25.

[8] X. Yin, Y. Zhu, J. Hu, A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions, ACM Comput. Surv. 54 (2021) 131:1–131:36, http://dx.doi.org/10.1145/3460427.

[9] C. Zhou, A. Fu, S. Yu, W. Yang, H. Wang, Y. Zhang, Privacy-preserving federated learning in fog computing, IEEE Internet Things J. 7 (2020) 10782–10793, http://dx.doi.org/10.1109/JIOT.2020.2987958.

[10] K. Wei, J. Li, M. Ding, C. Ma, H. Su, B. Zhang, H.V. Poor, User-level privacy-preserving federated learning: Analysis and performance optimization, IEEE Trans. Mob. Comput. 21 (2022) 3388–3401, http://dx.doi.org/10.1109/TMC.2021.3056991.

[11] V. Hartmann, K. Modi, J.M. Pujol, R. West, Privacy-preserving classification with secret vector machines, in: CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October (2020) 19-23, ACM, 2020, pp. 475–484, http://dx.doi.org/10.1145/3340531.3412051.

[12] J. Xu, B.S. Glicksberg, C. Su, P.B. Walker, J. Bian, F. Wang, Federated learning for healthcare informatics, J. Healthc. Informat. Res. 5 (2021) 1–19, http://dx.doi.org/10.1007/s41666-020-00082-4.

[13] P. Qiu, X. Zhang, S. Ji, T. Du, Y. Pu, J. Zhou, T. Wang, Your labels are selling you out: Relation leaks in vertical federated learning, IEEE Trans. Dependable Secure Comput. 20 (2023) http://dx.doi.org/10.1109/TDSC.2022.3208630.

[14] M. Gheisari, H.E. Najafabadi, J.A.A. Alzubi, J. Gao, G. Wang, A.A. Abbasi, A. Castiglione, OBPP: An ontology-based framework for privacy-preserving in IoT-based smart city, Future Gener. Comput. Syst. 123 (2021) 1–13, http://dx.doi.org/10.1016/j.future.2021.01.028.

[15] Y. Chen, Y. Su, M. Su, H. Chai, Y. Wei, S. Yu, FedTor: An anonymous framework of federated learning in internet of things, IEEE Internet Things J. 9 (2022) 18620–18631, http://dx.doi.org/10.1109/JIOT.2022.3162826.

[16] J. Domingo-Ferrer, A. Blanco-Justicia, J.A. Manjón, D. Sánchez, Secure and privacy-preserving federated learning via Co-utility, IEEE Internet Things J. 9 (2022) 3988–4000, http://dx.doi.org/10.1109/JIOT.2021.3102155.

[17] Y. Li, X. Tao, X. Zhang, J. Liu, J. Xu, Privacy-preserved federated learning for autonomous driving, IEEE Trans. Intell. Transp. Syst. 23 (2022) 8423–8434, http://dx.doi.org/10.1109/TITS.2021.3081560.

[18] L. Lyu, J. Yu, K. Nandakumar, Y. Li, X. Ma, J. Jin, H. Yu, K. Siong Ng, Towards fair and privacy-preserving federated deep models, IEEE Trans. Parallel Distrib. Syst. 31 (2020) 2524–2541, http://dx.doi.org/10.1109/TPDS.2020.2996273.

[19] A.M. Girgis, D. Data, S.N. Diggavi, P. Kairouz, A.T. Suresh, Shuffled model of federated learning: Privacy, accuracy and communication trade-offs, IEEE J. Sel. Areas Inf. Theory 2 (2021) 464–478, http://dx.doi.org/10.1109/JSAIT.2021.3056102.

[20] R. Dingledine, N. Mathewson, P.F. Syverson, Tor: The second-generation onion router, in: Proceedings of the 13th USENIX Security Symposium, August (2004) 9-13, San Diego, CA, USA, USENIX, 2004, pp. 303–320, http://www.usenix.org/publications/library/proceedings/sec04/tech/dingledine.html.

[21] Q. Tan, X. Wang, W. Shi, J. Tang, Z. Tian, An anonymity vulnerability in tor, IEEE/ACM Trans. Netw. 30 (2022) 2574–2587, http://dx.doi.org/10.1109/TNET.2022.3174003.

[22] L. Basyoni, N. Fetais, A. Erbad, A. Mohamed, M. Guizani, Traffic analysis attacks on tor: A survey, in: IEEE International Conference on Informatics, IoT, and Enabling Technologies, ICIoT 2020, Doha, Qatar, February (2020) 2-5, IEEE, 2020, pp. 183–188, http://dx.doi.org/10.1109/ICIoT48696.2020.9089497.

[23] S. Nepal, S. Dahal, S. Shin, Deanonymizing schemes of hidden services in tor network: A survey, in: 2015 International Conference on Information Networking, ICOIN 2015, Siem Reap, Cambodia, January (2015) 12-14, IEEE Computer Society, 2015, pp. 468–473, http://dx.doi.org/10.1109/ICOIN.2015.7057949.

[24] D.R. Cheriton, M. Gritter, Triad: A new next-generation internet architecture, 2000.

[25] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, R. Braynard, Networking named content, in: Proceedings of the 2009 ACM Conference on Emerging Networking Experiments and Technology, CoNEXT 2009, Rome, Italy, December (2009) 1-4, 2009, pp. 1–12, http://dx.doi.org/10.1145/1658939.1658941.

[26] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J.D. Thornton, D.K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, et al., Named data networking (NDN) project, Relatório Técnico NDN-0001 157 (2010) 158.

[27] M. Król, K. Habak, D. Oran, D. Kutscher, I. Psaras, RICE: remote method invocation in ICN, in: Proceedings of the 5th ACM Conference on Information-Centric Networking, ICN '18, Boston, Massachusetts, USA, September (2018) 21-23, ACM, 2018, pp. 1–11, http://dx.doi.org/10.1145/3267955.3267956.

[28] M. Król, S. Mastorakis, D. Oran, D. Kutscher, Compute first networking: Distributed computing meets ICN, in: Proceedings of the 6th ACM Conference on Information-Centric Networking, ICN 2019, Macao, SAR, China, September (2019) 24-26, ACM, 2019, pp. 67–77, http://dx.doi.org/10.1145/3357150.3357395.

[29] Y. Yu, A. Afanasyev, D. Clark, K. Claffy, V. Jacobson, L. Zhang, Schematizing trust in named data networking, in: Proceedings of the 2nd International Conference on Information-Centric Networking, ICN '15, San Francisco, California, USA, September 30 - October 2, 2015, ACM, 2015, pp. 177–186, http://dx.doi.org/10.1145/2810156.2810170.

[30] K. Nichols, Trust schemas and ICN: Key to secure home IoT, in: G. Carofiglio, D. Oran, J. Ott, L. Wang (Eds.), ICN '21: 8th ACM Conference on Information-Centric Networking, Paris, France, September (2021) 22-24, ACM, 2021, pp. 95–106, http://dx.doi.org/10.1145/3460417.3482972.

[31] H. Wang, M. Yurochkin, Y. Sun, D.S. Papailiopoulos, Y. Khazaeni, Federated learning with matched averaging, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April (2020) 26-30, OpenReview.net, 2020.

[32] L. Muñoz-González, K.T. Co, E.C. Lupu, Byzantine-robust federated machine learning through adaptive model averaging, CoRR abs/1909.05125, 2019.

[33] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 2017, Fort Lauderdale, FL, USA, in: Proceedings of Machine Learning Research, vol. 54, 2017, pp. 1273–1282, http://proceedings.mlr.press/v54/mcmahan17a.html.

[34] S.P. Karimireddy, S. Kale, M. Mohri, S.J. Reddi, S.U. Stich, A.T. Suresh, SCAFFOLD: Stochastic controlled averaging for federated learning, in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 2020, Virtual Event, in: Proceedings of Machine Learning Research, vol. 119, 2020, pp. 5132–5143, http://proceedings.mlr.press/v119/karimireddy20a.html.

[35] M. Safaryan, R. Islamov, X. Qian, P. Richtárik, FedNL: Making Newton-type methods applicable to federated learning, in: International Conference on Machine Learning, ICML 2022, 17-23 2022, Baltimore, Maryland, USA, in: Proceedings of Machine Learning Research, vol. 162, 2022, pp. 18959–19010, https://proceedings.mlr.press/v162/safaryan22a.html.

[36] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, Named data networking, Comput. Commun. Rev. 44 (2014) 66–73, http://dx.doi.org/10.1145/2656877.2656887.

[37] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, IEEE Commun. Mag. 50 (2012) 26–36, http://dx.doi.org/10.1109/MCOM.2012.6231276.

[38] W. Shang, A. Bannis, T. Liang, Z. Wang, Y. Yu, A. Afanasyev, J. Thompson, J. Burke, B. Zhang, L. Zhang, Named data networking of things (invited paper), in: First IEEE International Conference on Internet-of-Things Design and Implementation, IoTDI 2016, Berlin, Germany, April (2016) 4-8, IEEE Computer Society, 2016, pp. 117–128, http://dx.doi.org/10.1109/IoTDI.2015.44.

[39] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R.L. Aguiar, A.V. Vasilakos, Information-centric networking for the internet of things: Challenges and opportunities, IEEE Netw. 30 (2016) 92–100, http://dx.doi.org/10.1109/MNET.2016.7437030.

[40] A. Djama, B. Djamaa, M.R. Senouci, Information-centric networking solutions for the internet of things, Comput. Commun. 159 (2020) 37–59, http://dx.doi.org/10.1016/j.comcom.2020.05.003.

[41] H. Khelifi, S. Luo, B. Nour, H. Moungla, Y. Faheem, R. Hussain, A. Ksentini, Named data networking in vehicular ad hoc networks: State-of-the-art and challenges, IEEE Commun. Surv. Tutorials 22 (2020) 320–351, http://dx.doi.org/10.1109/COMST.2019.2894816.

[42] A. Boukerche, R.W.L. Coutinho, LoICen: A novel location-based and information-centric architecture for content distribution in vehicular networks, Ad Hoc Netw. 93 (2019) http://dx.doi.org/10.1016/j.adhoc.2019.101899.

[43] D.B. Rawat, R. Doku, A. Adebayo, C. Bajracharya, C.A. Kamhoua, Blockchain enabled named data networking for secure vehicle-to-everything communications, IEEE Netw. 34 (2020) 185–189, http://dx.doi.org/10.1109/MNET.001.1900593.

[44] G. Carofiglio, L. Muscariello, J. Augé, M. Papalini, M. Sardara, A. Compagno, Enabling ICN in the internet protocol: Analysis and evaluation of the hybrid-ICN architecture, in: Proceedings of the 6th ACM Conference on Information-Centric Networking, ICN 2019, Macao, SAR, China, September (2019) 24-26, ACM, 2019, pp. 55–66, http://dx.doi.org/10.1145/3357150.3357394.

[45] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, Found. Trends Theor. Comput. Sci. 9 (2014) 211–407, http://dx.doi.org/10.1561/0400000042.

[46] T. Zhu, P.S. Yu, Applying differential privacy mechanism in artificial intelligence, in: 39th IEEE International Conference on Distributed Computing Systems, ICDCS 2019, Dallas, TX, USA, July (2019) 7-10, 2019, pp. 1601–1609, http://dx.doi.org/10.1109/ICDCS.2019.00159.

[47] M. Hao, H. Li, G. Xu, S. Liu, H. Yang, Towards efficient and privacy-preserving federated deep learning, in: 2019 IEEE International Conference on Communications, ICC 2019, Shanghai, China, May (2019) 20-24, 2019, pp. 1–6, http://dx.doi.org/10.1109/ICC.2019.8761267.

[48] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, Y. Zhou, A hybrid approach to privacy-preserving federated learning, in: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2019, London, UK, November 15, 2019, 2019, pp. 1–11, http://dx.doi.org/10.1145/3338501.3357370.

[49] R. Hu, Y. Guo, H. Li, Q. Pei, Y. Gong, Personalized federated learning with differential privacy, IEEE Internet Things J. 7 (2020) 9530–9539, http://dx.doi.org/10.1109/JIOT.2020.2991416.

[50] K. Wei, J. Li, C. Ma, M. Ding, W. Chen, J. Wu, M. Tao, H.V. Poor, Personalized federated learning with differential privacy and convergence guarantee, IEEE Trans. Inf. Forensics Secur. 18 (2023) 4488–4503, http://dx.doi.org/10.1109/TIFS.2023.3293417.

[51] M. Seif, R. Tandon, M. Li, Wireless federated learning with local differential privacy, in: IEEE International Symposium on Information Theory, ISIT 2020, Los Angeles, CA, USA, June (2020) 21-26, IEEE, 2020, pp. 2604–2609, http://dx.doi.org/10.1109/ISIT44484.2020.9174426.

[52] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, K. Lam, Local differential privacy-based federated learning for internet of things, IEEE Internet Things J. 8 (2021) 8836–8853, http://dx.doi.org/10.1109/JIOT.2020.3037194.

[53] M. Adnan, S. Kalra, J.C. Cresswell, G.W. Taylor, H.R. Tizhoosh, Federated learning and differential privacy for medical image analysis, Sci. Rep. 12 (2022) 1953.

[54] A. El Ouadrhiri, A. Abdelhadi, Differential privacy for deep and federated learning: A survey, IEEE Access 10 (2022) 22359–22380, http://dx.doi.org/10.1109/ACCESS.2022.3151670.

[55] S. Shen, T. Zhu, D. Wu, W. Wang, W. Zhou, From distributed machine learning to federated learning: In the view of data privacy and security, Concurr. Comput.: Pract. Exper. 34 (2022) http://dx.doi.org/10.1002/cpe.6002.

[56] M. Shayan, C. Fung, C.J.M. Yoon, I. Beschastnikh, Biscotti: A blockchain system for private and secure federated learning, IEEE Trans. Parallel Distrib. Syst. 32 (2021) 1513–1525, http://dx.doi.org/10.1109/TPDS.2020.3044223.

[57] Y. Miao, Z. Liu, H. Li, K.R. Choo, R.H. Deng, Privacy-preserving Byzantine-robust federated learning via blockchain systems, IEEE Trans. Inf. Forensics Secur. 17 (2022) 2848–2861, http://dx.doi.org/10.1109/TIFS.2022.3196274.

[58] H. Kim, J. Park, M. Bennis, S. Kim, Blockchained on-device federated learning, Commun. Lett. 24 (2020) 1279–1283, http://dx.doi.org/10.1109/LCOMM.2019.2921755.

[59] S. Singh, S. Rathore, O. Alfarraj, A. Tolba, B. Yoon, A framework for privacy-preservation of IoT healthcare data using federated learning and blockchain technology, Future Gener. Comput. Syst. 129 (2022) 380–388, http://dx.doi.org/10.1016/j.future.2021.11.028.

[60] S.K. Singh, L.T. Yang, J.H. Park, FusionFedBlock: Fusion of blockchain and federated learning to preserve privacy in industry 5.0, Inf. Fusion 90 (2023) 233–240, http://dx.doi.org/10.1016/j.inffus.2022.09.027.

[61] J.A.A. Alzubi, O.A. Alzubi, A. Singh, M. Ramachandran, Cloud-iIoT-based electronic health record privacy-preserving by CNN and blockchain-enabled federated learning, Trans. Ind. Inf. 19 (2023) 1080–1087, http://dx.doi.org/10.1109/TII.2022.3189170.

[62] A. Dorri, S.S. Kanhere, R. Jurdak, P. Gauravaram, Blockchain for IoT security and privacy: The case study of a smart home, in: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops, 2017, Kona, Big Island, HI, USA, March (2017) 13-17, IEEE, 2017, pp. 618–623, http://dx.doi.org/10.1109/PERCOMW.2017.7917634.

[63] O.A. Alzubi, J.A.A. Alzubi, K. Shankar, D. Gupta, Blockchain and artificial intelligence enabled privacy-preserving medical data transmission in internet of things, Trans. Emerg. Telecommun. Technol. 32 (2021).

[64] D. Li, D. Han, T. Weng, Z. Zheng, H. Li, H. Liu, A. Castiglione, K. Li, Blockchain for federated learning toward secure distributed machine learning systems: A systemic survey, Soft Comput. 26 (2022) 4423–4440, http://dx.doi.org/10.1007/s00500-021-06496-5.

[65] D.C. Nguyen, M. Ding, Q. Pham, P.N. Pathirana, L.B. Le, A. Seneviratne, J. Li, D. Niyato, H.V. Poor, Federated learning meets blockchain in edge computing: Opportunities and challenges, IEEE Internet Things J. 8 (2021) 12806–12825, http://dx.doi.org/10.1109/JIOT.2021.3072611.

[66] S.A. Rahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, M. Guizani, A survey on federated learning: The journey from centralized to distributed on-site learning and beyond, IEEE Internet Things J. 8 (2021) 5476–5497, http://dx.doi.org/10.1109/JIOT.2020.3030072.

[67] S. DiBenedetto, P. Gasti, G. Tsudik, E. Uzun, Andana: Anonymous named data networking application, in: 19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February (2012) 5-8, The Internet Society, 2012, https://www.ndss-symposium.org/ndss2012/andana-anonymous-named-data-networking-application.

[68] K. Kita, Y. Koizumi, T. Hasegawa, O. Ascigil, I. Psaras, Producer anonymity based on onion routing in named data networking, IEEE Trans. Netw. Service Manag. 18 (2021) 2420–2436, http://dx.doi.org/10.1109/TNSM.2020.3019052.

[69] S.K. Ramani, R. Tourani, G. Torres, S. Misra, A. Afanasyev, NDN-ABS: Attribute-based signature scheme for named data networking, in: Proceedings of the 6th ACM Conference on Information-Centric Networking, ICN 2019, Macao, SAR, China, September (2019) 24-26, ACM, 2019, pp. 123–133, http://dx.doi.org/10.1145/3357150.3357393.

[70] J. Zhang, Q. Li, E.M. Schooler, iHEMS: An information-centric approach to secure home energy management, in: IEEE Third International Conference on Smart Grid Communications, SmartGridComm 2012, Tainan, Taiwan, November (2012) 5-8, IEEE, 2012, pp. 217–222, http://dx.doi.org/10.1109/SmartGridComm.2012.6485986.

[71] K. Nichols, Lessons learned building a secure network measurement framework using basic NDN, in: Proceedings of the 6th ACM Conference on Information-Centric Networking, ICN 2019, Macao, SAR, China, September (2019) 24-26, ACM, 2019, pp. 112–122, http://dx.doi.org/10.1145/3357150.3357397.

[72] A. Sapio, M. Canini, C. Ho, J. Nelson, P. Kalnis, C. Kim, A. Krishnamurthy, M. Moshref, D.R.K. Ports, P. Richtárik, Scaling distributed machine learning with in-network aggregation, in: 18th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2021, April (2021) 12-14, 2021, pp. 785–808, https://www.usenix.org/conference/nsdi21/presentation/sapio.

[73] C. Lao, Y. Le, K. Mahajan, Y. Chen, W. Wu, A. Akella, M.M. Swift, ATP: In-network aggregation for multi-tenant learning, in: 18th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2021, April (2021) 12-14, 2021, pp. 741–761, https://www.usenix.org/conference/nsdi21/presentation/lao.

[74] R. Viswanathan, A. Akella, Network-accelerated distributed machine learning using mlfabric, CoRR abs/1907.00434, 2019.

[75] E. Bardhi, M. Conti, R. Lazzeretti, E. Losiouk, Security and privacy of IP-ICN coexistence: A comprehensive survey, IEEE Commun. Surv. Tutor. (2023) 1, http://dx.doi.org/10.1109/COMST.2023.3295182.

[76] S. Mansoor, R. Patil, System and method for facilitating secure integration and communication of cloud services and enterprise applications, 2013, p. 609, US Patent 8, 504.

[77] S. Shailendra, B. Panigrahi, H.K. Rath, A. Simha, A novel overlay architecture for information centric networking, in: Twenty First National Conference on Communications, NCC 2015, Mumbai, India, February 27 - March 1, 2015, 2015, pp. 1–6, http://dx.doi.org/10.1109/NCC.2015.7084921.

[78] V. Mothukuri, R.M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, Future Gener. Comput. Syst. 115 (2021) 619–640, http://dx.doi.org/10.1016/j.future.2020.10.007.

[79] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, H. Ludwig, HybridAlpha: An efficient approach for privacy-preserving federated learning, in: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2019, London, UK, November 15, 2019, ACM, 2019, pp. 13–23, http://dx.doi.org/10.1145/3338501.3357371.

[80] G. Xu, H. Li, S. Liu, K. Yang, X. Lin, VerifyNet: Secure and verifiable federated learning, IEEE Trans. Inf. Forensics Secur. 15 (2020) 911–926, http://dx.doi.org/10.1109/TIFS.2019.2929409.

[81] J.R. Wallrabenstein, Practical and secure IoT device authentication using physical unclonable functions, in: 4th IEEE International Conference on Future Internet of Things and Cloud, FiCloud 2016, Vienna, Austria, August (2016) 22-24, IEEE Computer Society, 2016, pp. 99–106, http://dx.doi.org/10.1109/FiCloud.2016.22.

[82] R. Maes, I. Verbauwhede, Physically unclonable functions: A study on the state of the art and future research directions, in: Towards Hardware-Intrinsic Security - Foundations and Practice, Information Security and Cryptography, Springer, 2010, pp. 3–37, http://dx.doi.org/10.1007/978-3-642-14452-3_1.

[83] K.B. Frikken, M. Blanton, M.J. Atallah, Robust authentication using physically unclonable functions, in: Information Security, 12th International Conference, ISC 2009, Pisa, Italy, September (2009) 7-9. Proceedings, in: Lecture Notes in Computer Science, vol. 5735, Springer, 2009, pp. 262–277, http://dx.doi.org/10.1007/978-3-642-04474-8_22.

[84] H. Khelifi, S. Luo, B. Nour, H. Moungla, S.H. Ahmed, Reputation-based blockchain for secure NDN caching in vehicular networks, in: 2018 IEEE Conference on Standards for Communications and Networking,, CSCN 2018, Paris, France, October (2018) 29-31, 2018, pp. 1–6, http://dx.doi.org/10.1109/CSCN.2018.8581849.

[85] C. Fung, C.J.M. Yoon, I. Beschastnikh, Mitigating sybils in federated learning poisoning, CoRR abs/1808.04866, 2018.

[86] C. Briggs, Z. Fan, P. Andras, Federated learning with hierarchical clustering of local updates to improve training on non-IID data, in: International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July (2020) 19-24, IEEE, 2020, pp. 1–9, http://dx.doi.org/10.1109/IJCNN48605.2020.9207469.

[87] N. Dutta, An approach for FIB construction and interest packet forwarding in information centric network, Future Gener. Comput. Syst. 130 (2022) 269–278, http://dx.doi.org/10.1016/j.future.2022.01.005.

[88] G. Ács, M. Conti, P. Gasti, C. Ghali, G. Tsudik, Cache privacy in named-data networking, in: IEEE 33rd International Conference on Distributed Computing Systems, ICDCS 2013, 8-11 July, 2013, Philadelphia, Pennsylvania, USA, 2013, pp. 41–51, http://dx.doi.org/10.1109/ICDCS.2013.12.

[89] T. Chatterjee, S. Ruj, S.D. Bit, Security issues in named data networks, Computer 51 (2018) 66–75, http://dx.doi.org/10.1109/MC.2018.1151010.

[90] G. Ács, M. Conti, P. Gasti, C. Ghali, G. Tsudik, C.A. Wood, Privacy-aware caching in information-centric networking, IEEE Trans. Dependable Secure Comput. 16 (2019) 313–328, http://dx.doi.org/10.1109/TDSC.2017.2679711.

[91] E. Bardhi, M. Conti, R. Lazzeretti, E. Losiouk, ICN PATTA: ICN privacy attack through traffic analysis, in: 46th IEEE Conference on Local Computer Networks, Edmonton, AB, Canada, October (2021) 4-7, in: LCN 2021, 2021, pp. 443–446, http://dx.doi.org/10.1109/LCN52139.2021.9525013.

[92] J.R. Wallrabenstein, Practical and secure IoT device authentication using physical unclonable functions, in: M. Younas, I. Awan, W. Seah (Eds.), 4th IEEE International Conference on Future Internet of Things and Cloud, FiCloud 2016, Vienna, Austria, August (2016) 22-24, 2016, pp. 99–106, http://dx.doi.org/10.1109/FiCloud.2016.22.

[93] U. Guin, P. Cui, A. Skjellum, Ensuring proof-of-authenticity of IoT edge devices using blockchain technology, in: IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, July 30 - August 3, 2018, IEEE, 2018, pp. 1042–1049, http://dx.doi.org/10.1109/Cybermatics_2018.2018.00193.

[94] P. Cui, U. Guin, Countering botnet of things using blockchain-based authenticity framework, in: IEEE Computer Society Annual Symposium on VLSI, ISVLSI 2019, Miami, FL, USA, July (2019) 15-17, 2019, pp. 598–603, http://dx.doi.org/10.1109/ISVLSI.2019.00112.

[95] L. Negka, G. Gketsios, N.A. Anagnostopoulos, G.P. Spathoulas, A. Kakarountas, S. Katzenbeisser, Employing blockchain and physical unclonable functions for counterfeit IoT devices detection, in: Proceedings of the International Conference on Omni-Layer Intelligent Systems, COINS 2019, Crete, Greece, May (2019) 5-7, 2019, pp. 172–178, http://dx.doi.org/10.1145/3312614.3312650.

[96] E. Serpedin, Q.M. Chaudhari, Synchronization in Wireless Sensor Networks: Parameter Estimation, Performance Benchmarks, and Protocols, Cambridge University Press, 2009.

[97] V. Buhrmester, D. Münch, M. Arens, Analysis of explainers of black box deep neural networks for computer vision: A survey, Mach. Learn. Knowl. Extract. 3 (2021) 966–989, http://dx.doi.org/10.3390/make3040048.

[98] A. Agiollo, G. Ciatto, A. Omicini, Shallow2Deep: Restraining neural networks opacity through neural architecture search, in: Explainable and Transparent AI and Multi-Agent Systems. Third International Workshop, Vol. 12688, EXTRAAMAS 2021, Springer, 2021, pp. 63–82, http://dx.doi.org/10.1007/978-3-030-82017-6_5.

[99] A. Agiollo, A. Omicini, Load classification: A case study for applying neural networks in hyper-constrained embedded devices, Appl. Sci. 11 (2021) http://dx.doi.org/10.3390/app112411957.

[100] G. Jaume, P. Pati, B. Bozorgtabar, A. Foncubierta, A.M. Anniciello, F. Feroce, T. Rau, J. Thiran, M. Gabrani, O. Goksel, Quantifying explainers of graph neural networks in computational pathology, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 8106–8116, http://dx.doi.org/10.1109/CVPR46437.2021.00801.

[101] A. Agiollo, A. Omicini, GNN2gnn: Graph neural networks to generate neural networks, in: Uncertainty in Artificial Intelligence, in: Proceedings of Machine Learning Research, vol. 180, ML Research Press, 2022, pp. 32–42, https://proceedings.mlr.press/v180/agiollo22a.html.

[102] A. Agiollo, E. Bardhi, M. Conti, R. Lazzeretti, E. Losiouk, A. Omicini, GNN4ifa: Interest flooding attack detection with graph neural networks, in: 2023 IEEE 8th European Symposium on Security and Privacy (EuroS & P, IEEE Computer Society, 2023, pp. 615–630, http://dx.doi.org/10.1109/EuroSP57164.2023.00043.

[103] A. Agiollo, G. Ciatto, A. Omicini, Graph neural networks as the copula mundi between logic and machine learning: A roadmap, in: WOA 2021–22nd Workshop from Objects To Agents, in: CEUR Workshop Proceedings, vol. 2963, 2021, pp. 98–115, https://ceur-ws.org/Vol-2963/paper18.pdf.

[104] A. Agiollo, A. Rafanelli, A. Omicini, Towards quality-of-service metrics for symbolic knowledge injection, in: WOA 2022–23rd Workshop from Objects To Agents, Vol. 3261, 2022, pp. 30–47, https://ceur-ws.org/Vol-3261/paper3.pdf.

Andrea Agiollo received the Bachelor of Information Engineering degree from the University of Padua, Italy, in 2018, the first master's degree in information and communication technologies for Internet and multimedia from the University of Padua in 2020, and the second master's degree in communication engineering from National Taiwan University, Taiwan, in 2020. He is currently pursuing the Ph.D. degree in computer science and engineering with the University of Bologna, Italy, in collaboration with Electrolux Professional S.P.A. From October 2022 to February 2023, he was with Delft University of Technology in the Netherlands as a visiting Ph.D. student, and currently he is with Purdue University. His research interests include machine learning in resource-constrained environments, neuro-symbolic artificial intelligence, explainable artificial intelligence, and network security.

Enkeleda Bardhi received the Bachelor's Degree in Telecommunication Engineering in 2017 and the Master's Degree in ICT for Internet and Multimedia in 2020 from the University of Padua. She is currently pursuing the Ph.D. degree in Computer Science Engineering, focusing on Cybersecurity with Sapienza University of Rome. From October 2022 to February 2023, she was with Delft University of Technology in the Netherlands as a visiting Ph.D. student, and currently she is with Purdue University. Her research interests include network security and privacy and the application of Machine Learning on network security tasks.

Mauro Conti is Full Professor at the University of Padua, Italy. He is also affiliated with TU Delft and University of Washington, Seattle. He obtained his Ph.D. from Sapienza University of Rome, Italy, in 2009. After his Ph.D., he was a Post-Doc Researcher at Vrije Universiteit Amsterdam, The Netherlands. In 2011 he joined as Assistant Professor at the University of Padua, where he became Associate Professor in 2015, and Full Professor in 2018. He has been Visiting Researcher at GMU, UCLA, UCI, TU Darmstadt, UF, and FIU. He has been awarded with a Marie Curie Fellowship (2012) by the European Commission, and with a Fellowship by the German DAAD (2013). His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in the area of Security and Privacy. In this area, he published more than 500 papers in topmost international peer-reviewed journals and conferences. He is Editor-in-Chief for IEEE Transactions on Information Forensics and Security, Area Editor-in-Chief for IEEE Communications Surveys & Tutorials, and has been Associate Editor for several journals, including IEEE Communications Surveys & Tutorials, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, and IEEE Transactions on Network and Service Management. He was Program Chair for TRUST 2015, ICISS 2016, WiSec 2017, ACNS 2020, CANS 2021, CSS 2021, WiMob 2023 and ESORICS 2023, and General Chair for SecureComm 2012, SACMAT 2013, NSS 2021 and ACNS 2022. He is Fellow of the IEEE, Fellow of the AAIA, Senior Member of the ACM, and Fellow of the Young Academy of Europe.

Nicolò Dal Fabbro received the B.Sc. degree in Information Engineering and the M.Sc. degree in ICT for Internet and Multimedia from the University of Padova, Italy, in 2018 and 2020 respectively. He is currently pursuing the Ph.D. degree with the Department of Information Engineering at the University of Padova. His research interests are in machine learning over networks, edge computing, and wireless sensing.

Riccardo Lazzeretti received the Computer Science Engineering degree (cum laude) and the Ph.D. degree from the University of Siena, Italy, in 2007 and 2012, respectively. From November 2009 to May 2010, he was with Philips Lab, Eindhoven, The Netherlands From 2012 to 2015, he continued his research with the University of Siena, and from 2016 to March 2017, with the University of Padua, Italy. Since 2022, he is an Associate Professor with the Department of Computer, Control, and Management Engineering at Sapienza University of Rome, Italy. His research activity mainly focuses on security and privacy. He is an IEEE Senior Member, an elected member of the IEEE Information Forensics and Security Technical Committee, and an Associate Editor of the Journal of Information Security and Applications (Elsevier).