# Topology-Based Reconstruction Prevention for Decentralised Learning

Florine W. Dekker
Delft University of Technology
Delft, Netherlands
f.w.dekker@tudelft.nl

Zekeriya Erkin
Delft University of Technology
Delft, Netherlands
z.erkin@tudelft.nl

Mauro Conti
Università di Padova
Padua, Italy
Delft University of Technology
Delft, Netherlands
mauro.conti@unipd.it

## Abstract

Decentralised learning has recently gained traction as an alternative to federated learning in which both data and coordination are distributed over its users. To preserve the confidentiality of users' data, decentralised learning relies on differential privacy, multi-party computation, or a combination thereof. However, running multiple privacy-preserving summations in sequence may allow adversaries to perform reconstruction attacks. Unfortunately, current reconstruction countermeasures either cannot trivially be adapted to the distributed setting, or add excessive amounts of noise.

In this work, we first show that passive honest-but-curious adversaries can infer other users' private data after several privacy-preserving summations. For example, in subgraphs with 18 users, we show that only three passive honest-but-curious adversaries succeed at reconstructing private data 11.0% of the time, requiring an average of 8.8 summations per adversary. The success rate depends only on the adversaries' direct neighbourhood, and is independent of the size of the full network. We consider weak adversaries that do not control the graph topology, cannot exploit the inner workings of the summation protocol, and do not have auxiliary knowledge; and show that these adversaries can still infer private data.

We develop a mathematical understanding of how reconstruction relates to topology and propose the first topology-based decentralised defence against reconstruction attacks. Specifically, we show that reconstruction requires a number of adversaries linear in the length of the network's shortest cycle. Consequently, exact reconstruction attacks over privacy-preserving summations are impossible in acyclic networks.

Our work is a stepping stone for a formal theory of topology-based decentralised reconstruction defences. Such a theory would generalise our countermeasure beyond summation, define confidentiality in terms of entropy, and describe the interactions with (topology-aware) differential privacy.

## Keywords

reconstruction attacks, statistical disclosure, decentralised learning, privacy-preserving summation, graph girth

## 1 Introduction

Machine learning is used in a wide array of systems, including smartwatches [52], predictive text [6], and malware detection [43]. These systems require access to large amounts of reliable data in order to function accurately. In practice, the necessary data usually exist, but are distributed over many data owners. The naive approach for data collection is to have the data owners send their data to a central server, which trains a machine learning model on these data before deploying it. However, sharing private data may result in misuse, for example in the form of targeted advertising or harassment. In an age of increasing privacy awareness, data owners may be reluctant to share their data, threatening the viability of data-intensive machine learning applications.

The emerging field of federated learning, first formalised in [40], addresses these privacy issues by distributing the training process over the data owners. Instead of submitting their data, each data owner first trains a machine learning model on their local data and then submits this model to a central server. This central server, called the aggregator, uses a privacy-preserving summation protocol to combine the received models into a single global model. The central server then sends back the global model to the data owners, who apply another round of training, repeating the entire process until the global model has converged.

A significant drawback of classical federated learning is that communication is a bottleneck, scaling quadratically [6] or poly-logarithmically [2] in the number of users. Decentralised learning, a variant of federated learning [33], removes this bottleneck by distributing both the data and the coordination between users. Training happens in a peer-to-peer fashion, with users exchanging information only with their direct neighbours. This significantly reduces the communication complexity [37], allowing for cost-effective deployments without a central server. Furthermore, because communication is local, it becomes much harder for adversaries to observe the full network [47].

Recently, there has been increased interest in decentralised learning. Though some works do not consider privacy [37, 45, 55], many other works do. Some of these works [3, 48, 55] consider algorithms in which nodes are randomly selected to calculate updates, and protect the private data underlying the models using differential privacy. That is, they apply carefully calibrated random noise to the calculated gradients before sharing them with others. A slight variation of this is to use a random walk through the graph to determine the order in which updates occur [17]. There are also works [10, 42, 44] that use blockchains to facilitate the communication and coordination between nodes, and then similarly use

*Proceedings on Privacy Enhancing Technologies 2025(1), 553–566*

https://doi.org/10.56553/popets-2025-0030

differential privacy. Finally, instead of differential privacy, some works utilise multi-party computation [20, 34, 46], which does not give noisy results, but has higher computational costs.

A common thread in these works is that they apparently assume that if a single summation is secure, then the protocol remains secure after multiple summations. However, this requires further scrutiny, as combining information from multiple rounds may reveal previously hidden information. For example, given private records $A$, $B$, and $C$, and a privacy-preserving summation protocol, an adversary could separately query $A + B$, then $B + C$, and finally $A + C$, and use a linear algebra solver to learn all three private records. To defend against such attacks, one must prevent sequences of queries that would reveal private data. Naive restrictions, such as requiring a minimum number of included records per query, are insufficient: The adversary could still first query the sum of all models and then query the sum of all models except one, allowing them to reconstruct the excluded model. As such, designing proper countermeasures requires a formal theory.

Extracting data from output traces is known as a reconstruction attack, which has its roots in the theory of statistical disclosure [28]. Many defences have been proposed since the 1970s, including query auditing [13], perturbation [25], and random sampling [23]. However, these works assume either a central database, or otherwise assume a central arbiter that determines which queries are allowed. In decentralised learning, there is no clear leader who can be trusted to audit queries. Instead, decentralised learning requires a decentralised solution. Apart from works on perturbation, to the best of our knowledge, only da Silva et al. [18] have considered reconstruction attacks in peer-to-peer networks, but their work applies only to distributed clustering, and does not propose any countermeasures. When considering perturbation, naively applying user-level differential privacy in a distributed setting results in linearly-scaling noise, severely reducing the protocol's utility [17, 26, 56]. Intuitively, utility can be increased while retaining the level of privacy by correlating noise by topology [25], but to the best of our knowledge only a few works have done this. Guo et al. [30] reduce noise based on the mutual overlaps of neighbours' neighbourhoods, but do not consider time-series correlations. Cyffers et al. [17] observe that data sensitivity decreases as mutual node distances increase, but their solution does not scale well under collusion.

In this work, we analyse reconstruction attacks performed by colluding adversaries in peer-to-peer networks. We model the network after decentralised learning, though our analysis is sufficiently generic to describe a sequence of summations in any environment. Summation is a simple protocol, but is sufficient to implement many of the aforementioned decentralised learning protocols, in addition to smart metering [29] and even principal component analysis, singular-value decomposition, and decision tree classifications [4]. We assume a set of nodes, each with a private datum that changes over time, and allow privacy-preserving summation over one's direct neighbours. We do not consider auxiliary knowledge; see Section 3.3.3 and [14, 15] for a detailed discussion on the real-world applicability of this model. We then formalise the relation between reconstruction and network topology, and prove that exact reconstruction attacks are impossible in a specific class of topologies.

Concretely, we begin by showing that reconstruction attacks are practical, and that, in random peer-to-peer subgraphs, three honest-but-curious adversaries with 15 neighbours succeed in finding at least one neighbours' private datum with an 11.0% success rate, requiring an average of only 8.8 rounds per adversary. The success rate is independent of the size of the full network; it depends only on the adversaries' local neighbourhood. We then show that the success rate depends on the connectivity of the network rather than its size. Specifically, we show that reconstruction corresponds to cycles in the graph: If the graph's shortest cycle has length $2k$, then reconstruction never succeeds if there are fewer than $k$ adversaries. Finally, we briefly evaluate the impact of increasing girth on the convergence of a distributed averaging protocol, and find that while more rounds are required to achieve convergence in all graphs, dense graphs require fewer rounds than sparse graphs do when both are "stretched" to higher girths.

To the best of our knowledge, our work is the first to propose a topology-based decentralised defence to reconstruction attacks. We show that restricting how summations may be composed makes it impossible to reconstruct private data. We must therefore assume that adversaries do not have auxiliary knowledge, as restrictions on summations cannot be guaranteed otherwise. With the ultimate goal of developing a general theory of structured composition as a distributed reconstruction countermeasure, future work may include finding a condition that is not only sufficient (as seen in this work) but also necessary for reconstruction, generalising these countermeasures to operations beyond summation, stronger notions of privacy rooted in information theory, and investigating the interactions with (topology-aware) differentially private noise.

The remainder of this paper is structured as follows. In Section 2, we discuss related work. In Section 3, we describe the preliminaries: We explain basic primitives, formalise our assumptions, and introduce our notation. In Section 4, we formally describe reconstruction attacks, and show that the attack is feasible. In Section 5, we prove that the success rate of the reconstruction attack depends on the graph's girth, and investigate how girth affects application performance. Finally, in Section 6, we present our conclusions.

## 2 Related Work

In this work we propose a decentralised reconstruction countermeasure for privacy-preserving summation with dynamic data. To the best of our knowledge, this exact problem has not been treated in literature before. Therefore, in this section, we consider related works from various fields, and describe their similarities and differences.

### 2.1 Reconstruction Attacks

Consider a database that users can query for statistical information. For example, in a database with employee records, users can query for the sum of salaries of all PhD students. Naturally, the database must ensure that users cannot learn individual employees' salaries. A naive defence would disallow queries over single records, but a cleverly chosen sequence of queries may still allow the user to reconstruct private data. For example, the user could query the sum of salaries of all employees, and the sum of salaries of all employees except Jay Doe, and reconstruct Jay Doe's salary from that.

The attack described above is known under various names: *statistical disclosure*[1], the *inference problem*, and the *reconstruction attack*. It has been the subject of research since at least the 1970s [28], originally in the context of releasing census statistics. Since then, many reconstruction defences have been proposed, including random sampling [23], query auditing [13], and perturbation [25].

Most related to our research question are those works that consider sum queries only. Chin [12] studies summation query graphs to determine the exact conditions under which disclosure occurs. However, his analysis is limited to queries that are over exactly two records each, and cannot easily be generalised. Wang et al. [49] allow queries over more than two records. The authors propose cardinality-based criteria for determining whether reconstruction is possible, and create a whitelist of all summations that can be performed without causing undesired reconstruction.

All aforementioned solutions consider a single trusted database or auditor, making them unsuitable for peer-to-peer protocols, in which the data are spread over many users. Except for perturbation-based techniques, there are very few works that consider reconstruction defences in peer-to-peer settings. In their study on reconstruction attacks in distributed environments, Jebali et al. [32] note only the work by da Silva et al. [18] when discussing peer-to-peer solutions, but the latter applies only to distributed clustering, and does not propose any countermeasures.

Perturbation, on the other hand, has been studied in more detail. Probably the most popular perturbation mechanism for the decentralised setting is local differential privacy [27, 35, 51], a variation of differential privacy [25]. With this technique, when a query is performed over some set of nodes, each node adds a small amount of noise such that the aggregate is relatively accurate, but reconstruction remains impossible even after multiple queries. Various fully-decentralised learning protocols use local differential privacy to allow learning a shared machine learning model without revealing users' private datasets [3, 48, 55]. However, the perturbation is calibrated to protect individual records in users' private datasets, rather than protecting users' entire datasets. As a result, these works are potentially vulnerable to inversion attacks [31, 50]. The level of noise can be increased, but this severely impacts utility [17, 56]. Intuitively, noise can be made more "efficient" by exploiting correlations between users' data [26], which, in peer-to-peer networks, amounts to calibrating noise to the topology. To the best of our knowledge only a few works have done this. Guo et al. [30] reduce noise based on the mutual overlaps of neighbours' neighbourhoods, but do not consider time-series correlations. Cyffers et al. [17] observe that data sensitivity decreases as mutual node distances increase, but their solution does not scale well when adversaries collude.

## 2.2 Multi-Party Computation

In secure multi-party computation, composability [38] is the property of a cryptographic scheme that no additional leakage occurs when it is invoked multiple times, with varying parties, combined with other schemes, and so on. There are numerous frameworks to model composability, including universal composability [8], constructive composability [39], and reactive simulatability [1].

Composability solves a different issue than the one posed in this work. While composability ensures nothing leaks beyond what can be inferred from the outputs, our work is concerned exactly with that which can be inferred from the outputs. Composability does not help when the desired output (implicitly) reveals private data.

In secure multi-party computation literature, this difference is occasionally acknowledged. For example, Bogdanov et al. [5] note that "the composition of ideal functionalities is no longer an ideal functionality", and, before them, Yang et al. [54] made a similar observation. There are more works that consider this difference, but, to the best of our knowledge, these works all resolve the issue by removing or protecting intermediate values, but do not consider protocols which desire intermediate values, and even then do not consider that reconstruction attacks may be possible after multiple instantiations of the protocol. An exception is the work by Dekker and Erkin [21], which releases intermediate values in a structured manner such that it is not possible to reconstruct all users' values. However, the authors do not prove (or disprove) that it is impossible to find a *single* user's value.

# 3 Preliminaries

We briefly explain some basics on privacy-preserving summation in Section 3.1 and on bipartite graphs in Section 3.2. After that, we formulate our assumptions and define our notation in Section 3.3.

## 3.1 Privacy-Preserving Summation

Privacy-preserving summation is a special case of multi-party computation in which an aggregator calculates the sum of users' private values without learning the users' individual values. In this work, we consider privacy-preserving summation to be an information-theoretically secure black-box that reveals only the identities and the sum of the variables.

## 3.2 Bipartite Graphs

A bipartite graph $H = (U, V, E)$ is a graph with nodes $U \cup V$ and edges $E$, subject to $U \cap V = \emptyset$ and $\forall (u, v) \in E : u \in U \Leftrightarrow v \in V$.

Furthermore, a bipartite graph $H = (U, V, E)$ can be described by a biadjacency matrix $A \in \{0, 1\}^{|U| \times |V|}$, where $\forall 0 \leq u < |U|, 0 \leq v < |V| : A_{u,v} = 1 \Leftrightarrow (U_u, V_v) \in E$.

In this work, all graphs are undirected.

## 3.3 Assumptions and Notation

The underlying models and assumptions in this work are based on those seen in the decentralised learning literature [3, 20, 55], but are especially close to the work by Vanhaesebrouck et al. [48].

In general, we denote the first element of a vector $v$ by $v_0$, the first row of a matrix $A$ by $A_0$, the range of integers $\{0 \ldots n - 1\}$ by $[\![n]\!]$, and the number of elements in a collection $S$ by $|S|$.

*3.3.1 User data and objectives.* Consider a system of $n$ users $V$, each with a private datum. Each datum is dynamic; it changes each time the user initiates a round and incorporates new knowledge from their neighbours. (We describe the time model in Section 3.3.4.) Each datum can be a vector of values, though for simplicity we

[1] Confusingly, the term "statistical disclosure attack" is also a separate attack in peer-to-peer literature [19], but this is an unrelated attack on anonymity rather than confidentiality.

assume scalar values in our notation. Examples of dynamic data are power consumption, GPS coordinates, and machine learning models. In round $t$, the data of user $i \in [\![n]\!]$ is denoted $\theta_{i,t}$.

The users want to compute some function over their data without revealing their data to others. Each user regularly runs a privacy-preserving summation protocol to find the sum of their direct neighbours' private data. This sum can be used for principal component analysis, singular-value decomposition, or distributed gradient descent, for example.

*3.3.2 Network model.* Users communicate with each other in a peer-to-peer network. This can be a physical network, for example based on Bluetooth or Wi-Fi Direct, or an overlay network, in which users are connected through the Internet. We model the network as an undirected, self-loopless, static graph $G = (V, E)$ in which each node represents a user. (We consider graphs with dynamic edges in Section 5.4.) The direct neighbours of a node $v \in V$ are denoted $N_G(v)$, and for any set of users $U \subseteq V$ we define their shared neighbours $N_G(U) \coloneqq \bigcup_{u \in U} N_G(u) \setminus U$. The network topology is not private; in fact, users know who their direct neighbours are. Users may run a privacy-preserving summation protocol to learn the sum of their direct neighbours' private values.

*3.3.3 Adversarial model.* We assume all $n$ users $V$ are honest-but-curious. That is, all users honestly follow the protocol, but may attempt to obtain other users' private data by operating on the data obtained in the protocol in any way they see fit. Additionally, $k$ users $C \subseteq V$ may collude with each other, but we require that each adversary has either zero or at least two non-adversary neighbours, as retrieving private data is trivial otherwise. We give an example of a valid set of adversaries in Figure 1. Colluding users are still honest-but-curious, so their collusion is limited to sharing information outside the protocol. While excluding all actively malicious behaviour is a strenuous assumption in practice, we argue that the challenges in the honest-but-curious model are already sufficiently interesting to warrant investigation. We leave stronger notions of adversarial behaviour to future work; see also Section 6.

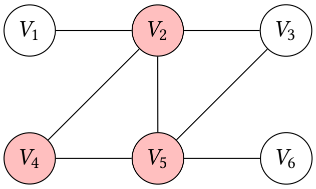


**Figure 1: A network with 6 users $V$. The adversaries $C = \{V_2, V_4, V_5\}$ are shaded. Removing edge $(V_2, V_3)$ would violate our requirements, as adversary $V_2$ would have exactly one non-adversary neighbour.**

Finally, we assume that adversaries do not possess auxiliary knowledge. That is, we aim for syntactic privacy [14], of which the privacy guarantees do not compose trivially with those of other protocols using the same private data. Syntactic privacy is suitable when high utility is desired and participants have some level of mutual trust [14, 15]. Moreover, prescribing a syntax on the data is inherent to this work's goal of establishing an interpretable relation between privacy and topology. We note that syntactic privacy does not preclude the use of semantic protections such as differential privacy, though the investigation of that combination is out of scope for this work. See [14, 15] for a detailed discussion of the subject.

*3.3.4 Time model.* We work in the asynchronous time model [7], in which a global clock ticks whenever a user wakes up and performs some work. Equivalently, each user has their own clock ticking at the speed of a rate-1 Poisson process; when a user's clock ticks, that user wakes up. We denote the current global round number by $t$ (for "time").

## 4 Reconstruction in Multi-party Summation

In this section we formally define reconstruction attacks in privacy-preserving multi-party dynamic-data summation, and experimentally verify that this attack is feasible. Adversaries passively record the summations they obtain throughout the protocol. Because adversaries know which users are included in which summation, they obtain a system of linear equations. Even if the system has no global solutions, adversaries may still learn the private data of some users.

In Section 4.1, we informally explain reconstruction attacks with examples. In Section 4.2, we give an exact definition of the adversaries' knowledge. In Section 4.3, we formally define reconstruction on multi-party dynamic-data summation. In Section 4.4, we experimentally verify the feasibility and success rate of reconstruction attacks on random graphs.

### 4.1 Introduction to Reconstruction Attacks

For this brief introduction, we use somewhat informal notation. We formally define our notation in Section 4.2.

*A small example.* Consider a graph $G = (V, E)$ with users $V$ and a set of $k$ adversaries $C \subseteq V$. If a single adversary $c \in C$ sums their neighbours' values, they learn a linear equation $\Theta_c$ over the private values $\theta$ of neighbours $N_G(c)$. If multiple adversaries $C$ collude, they share a *system* of linear equations $A\theta = \Theta$ over the private values $\theta$ of $N_G(C)$. If the system of linear equations has a solution, then the adversaries are able to calculate all observed users' private values using linear combinations of the system's rows. For example, given adversaries $A$, $B$, and $C$ with observations

$$\begin{array}{ccccccc} \theta_1 & + & \theta_2 & & & = & \Theta_A, \\ \theta_1 & & & + & \theta_3 & = & \Theta_B, \text{ and} \\ & & \theta_2 & + & \theta_3 & = & \Theta_C, \end{array} \tag{1}$$

this is equivalent to the system of linear equations

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \theta = \Theta. \tag{2}$$

Since this system is full rank, adversaries can calculate

$$\theta_1 = \frac{\Theta_A + \Theta_B + \Theta_C}{2} - \Theta_C, \tag{3}$$

$$\theta_2 = \Theta_A - \theta_1, \text{ and} \tag{4}$$

$$\theta_3 = \Theta_B - \theta_1. \tag{5}$$

For example, if $\Theta_A = 7$, $\Theta_B = 13$, and $\Theta_C = 8$, the adversaries know with certainty that $\theta_1 = 6$, $\theta_2 = 1$, and $\theta_3 = 7$. Observe that this works even if each individual summation in Equation 1 is information-theoretically secure.

*Partial solutions.* If the system is rank-deficient, no unique solution exists, but the system may still have partial solutions. That is, even if a system has infinitely many possible solutions, it may be the case that some variables have the same value in all solutions. Even a single user's private value being leaked is a major issue for any privacy-preserving protocol. Consider, for example, the adversarial knowledge consisting of

$$\begin{array}{ccccccl} \theta_1 & + & \theta_2 & + & \theta_3 & = & \Theta_A \text{ and} \\ \theta_1 & + & \theta_2 & & & = & \Theta_B. \end{array} \tag{6}$$

Even though there is no unique solution, all solutions have the same value for $\theta_3$, calculated as $\theta_3 = \Theta_A - \Theta_B$.

The case of Equation 6 is trivial because $\Theta_B$ is the sum over a subset of $\Theta_A$. However, there are also rank-deficient systems in which no summation is a subset of another:

$$\begin{array}{cccccccl} \theta_1 & + & \theta_2 & + & \theta_3 & & & = \Theta_A, \\ \theta_1 & + & \theta_2 & & & + & \theta_4 & = \Theta_B, \text{ and} \\ & & & & \theta_3 & + & \theta_4 & = \Theta_C. \end{array} \tag{7}$$

This system, too, has an infinite number of solutions, but each possible solution has the same values

$$\theta_3 = \frac{\Theta_A + \Theta_C - \Theta_B}{2} \text{ and} \tag{8}$$

$$\theta_4 = \frac{\Theta_B + \Theta_C - \Theta_A}{2}. \tag{9}$$

*Time dimension.* The above examples do not take into account that users' data change over time. To model dynamic data, first recall from Section 3.3.1 that users update their values only after initiating a summation. Since each update requires an interactive summation, users implicitly inform their neighbours whenever they update; and since each update represents the introduction of a new unknown value to $\theta$, adversaries can represent an update by adding a new column to their adversarial knowledge. If a user updates their value multiple times before being observed by an adversary, the adversaries treat this as a single update.

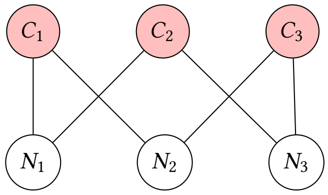


**Figure 2: Example graph $G$ with adversaries $C = \{C_1, C_2, C_3\}$ (shaded) and non-adversaries $N = N_G(C) = \{N_1, N_2, N_3\}$.**

To give an example, consider adversaries $C$ and their neighbours $N_G(C)$ in Figure 2. Say that initially adversaries $C_1$ and $C_2$ run their summations, learning

$$\left[\begin{array}{c|c|c} 1 & 1 & 0 \\ 1 & 0 & 1 \end{array}\right]\theta = \Theta. \tag{10}$$

The added vertical lines group the columns per non-adversarial user. Next, say that user $N_1$ updates their private value. This is noticed by the adversaries, who insert a new column into their system of equations. If user $C_1$ then does another summation (which includes user $N_1$'s new value), the adversaries know

$$\begin{array}{l} \;\;\overbrace{\hphantom{1\quad 0}}^{N_1} \\ \left[\begin{array}{cc|c|c} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right]\theta = \Theta. \end{array} \tag{11}$$

The last row represents adversary $C_1$'s new summation, and the second column represents user $N_1$'s new value. Finally, if users $N_1$, $N_2$, and $C_1$ subsequently update (in that order), then users $N_1$ and $N_2$ each get a new column, and $C_1$'s update adds a new row, giving

$$\begin{array}{l} \;\;\overbrace{\hphantom{1\quad 0\quad 0}}^{N_1}\;\overbrace{\hphantom{1\quad 0}}^{N_2} \\ \left[\begin{array}{ccc|cc|c} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{array}\right]\theta = \Theta. \end{array} \tag{12}$$

In the remainder of this work, to simplify notation, we will always assign the same number of columns $t$ to each user.

*Observations.* Before we give a formal definition of reconstruction attacks, we make two observations:

1. Reconstruction does not rely on weaknesses in the summation algorithm; **reconstruction works even if summation is done by a trusted third party**. Instead, reconstruction relies only on the summation revealing both the identities of included variables and the sum of those variables.
2. Reconstruction is independent of how users update their private values, and works even if users update their models in random ways or multiple times. **Reconstruction works because adversaries observe multiple summations with at least one unchanged value, and know how the summations are related.**

## 4.2 Obtained Adversarial Knowledge

We give a formal description of adversarial knowledge, which is the system of linear equations that adversaries obtain in a privacy-preserving multi-party dynamic-data summation protocol, and observe two important properties.

Let $G = (V, E)$ be an undirected graph, let $C \subseteq V$ be a collusion of $k$ adversaries, let $n \coloneqq |N_G(C)|$, and let $t \in \mathbb{N}$ be the number of summations performed by $C$.

**Definition 1** (Adversarial knowledge)**.** The adversarial knowledge over $t$ summations by $C$ is a consistent system of linear equations $A\theta = \Theta$, subject to the conditions that

- $\theta \in \mathbb{R}^{nt \times 1}$ are the private values of neighbours $N_G(C)$, such that $\theta_{vt+i}$ is the $i \in [\![t]\!]$th unique private value of neighbour $v \in [\![n]\!]$ that is observed by any adversary in $C$,
- $\Theta \in \mathbb{R}^{t \times 1}$ are the sums obtained by the adversaries, where $\Theta_\tau$ is the $\tau \in [\![t]\!]$th such sum, and
- $A \in \{0,1\}^{t \times nt}$ indicates which private values are observed in which summation, such that $A_{\tau, vt+i} = 1$ if and only if the adversaries' $\tau \in [\![t]\!]$th summation includes the $i \in [\![t]\!]$th unique private value of neighbour $v \in [\![n]\!]$.

*Remark* 1. In Theorem 2, we will show that it is not necessary to include adversaries' own private values in $A\theta = \Theta$.

**Property 1.** Let $A$ be the adversarial knowledge over $t$ summations by $C$. In each equation, each neighbour in $N_G(C)$ contributes at most one private value:

$$\forall \tau \in [\![ t ]\!], v \in [\![ n ]\!] : \sum_{i \in [\![ t ]\!]} A_{\tau, vt+i} \in \{0, 1\}. \tag{13}$$

**Property 2.** Let $A$ be the adversarial knowledge over $t$ summations by $C$. Since each equation is over all the neighbours of an adversary in $C$, each row in $A$ corresponds exactly to $N_G(c)$ for some $c \in C$:

$$\forall \tau \in [\![ t ]\!] : \exists c \in C : \forall v \in [\![ n ]\!] : \left( \sum_{i \in [\![ t ]\!]} A_{\tau, vt+i} = 1 \right) \Leftrightarrow (c, N_G(C)_v) \in E. \tag{14}$$

As in Property 1, the summation merely describes whether neighbour $v$ is included in the $\tau$th linear equation.

### 4.3 Reconstruction from Adversarial Knowledge

Finding a (partial) solution is not trivial. It is well-known that the reduced row echelon form (rref) of a system of linear equations reveals the system's unique solution, if it has one. Clearly, this unique solution is also at least a partial solution. However, if there is no unique solution, there may still be a partial solution, as in Equation 6. We will show in Theorem 1 that finding the reduced row echelon form of the adversarial knowledge is both necessary and sufficient to find all partial solutions. Moreover, we will show in Theorem 2 that this is true even if adversaries' own private values are removed from the adversarial knowledge matrix.

We begin with some definitions. Let $G = (V, E)$ be an undirected graph, let $C \subseteq V$ be a set of $k$ adversaries, let $n := |N_G(C)|$, let $t \in \mathbb{N}$, and let $A\theta = \Theta$ be the adversarial knowledge over $t$ summations by $C$; that is, $A \in \mathbb{R}^{t \times nt}$.

**Definition 2** (Solution of a variable)**.** Let $y \in \mathbb{R}^{1 \times t}$ and let $i \in [\![ nt ]\!]$. We say that "$y$ solves $\theta_i$ in $A\theta = \Theta$" if and only if the vector $yA$ contains exactly one non-zero value, at index $i$:

$$\left( (yA)_i \neq 0 \right) \wedge \left( \forall j \in [\![ nt ]\!] \setminus i : (yA)_j = 0 \right). \tag{15}$$

*Remark* 2. Since Equation 15 is independent of $\theta$ and $\Theta$, it is equivalent to say that "$y$ solves $\theta_i$ in $A$".

**Definition 3** (Partial solution)**.** Let $y \in \mathbb{R}^{1 \times t}$. If $y$ solves $\theta_i$ in $A$ for any $i \in [\![ nt ]\!]$, then we say that "$y$ is a partial solution to $A$".

We proceed with the central theorem of this section, which states that the reduced row echelon form of $A$ describes all partial solutions to $A$. We remark that a weaker variant of this theorem is given by Wang et al. [49] without a formal proof.

**Theorem 1.** *Let $i \in [\![ nt ]\!]$, and let $B \in \mathbb{R}^{t \times t}$ such that $BA = \text{rref}(A)$. Then $\theta_i$ has a solution in $A$ if and only if there exists $r \in [\![ t ]\!]$ such that $B_r$ solves $\theta_i$ in $A$.*

Proof. Given $i \in [\![ nt ]\!]$, we give a proof for both directions.

We first prove that if there exists $r \in [\![ t ]\!]$ such that $B_r$ solves $\theta_i$, then $\theta_i$ has a solution in $A$. Since $A\theta = \Theta$, it follows that $B_r A\theta = B_r\Theta$, and by Equation 15 we have that $B_r A\theta = \theta_i$. Therefore, $\theta_i = B_r\Theta$. This proves the first direction of Theorem 1.

We prove the other direction of Theorem 1 by contradiction. Let $y \in \mathbb{R}^{1 \times t}$ be a solution to $\theta_i$ in $A$, so $yA$ has its only non-zero value at $(yA)_i$. For the sake of contradiction, assume that there is no row in $B$ that solves $\theta_i$ in $A$. Because $y$ is in the row space of $A$, and the row space of $A$ is the same as the row space of $\text{rref}(A)$, there exists $y' \in \mathbb{R}^{1 \times t}$ such that $yA = y' \cdot \text{rref}(A) = y'BA$. By associativity of matrix multiplication, $y'B$ solves $\theta_i$ in $A$. Furthermore, since we assumed (for the sake of contradiction) that no single row of $B$ solves $\theta_i$ in $A$, it follows that $y'$ must have multiple non-zero coefficients. Thus, let $y'_r$ and $y'_s$ be any two non-zero coefficients in $y'$, and let $j, k$ such that $(BA)_{r,j}$ and $(BA)_{s,k}$ are the leading coefficients of their respective rows; these are their columns' only non-zero values, and $j \neq k$. Therefore, $(yA)_j = (y'BA)_j = y'_r \neq 0$, and similarly $(yA)_k = y'_s \neq 0$. However, this is a contradiction, because we initially assumed that $yA$ has its only non-zero value at $(yA)_i$. Therefore, there exists a row in $B$ that solves $\theta_i$ in $A$. This proves the other direction of Theorem 1.

Therefore, it is both necessary and sufficient to check the rows of $BA = \text{rref}(A)$ to learn all partial solutions to $A$. □

Note that $A$ does not describe that adversaries know each other's private values, since $N_G(C)$ excludes adversaries themselves. We show that including this knowledge does not reveal new partial solutions. Specifically, observe that the adversarial knowledge *including self-knowledge* over $t$ summations by $k$ adversaries $C$ is

$$A' = \begin{bmatrix} A & R \\ 0 & I_{tk} \end{bmatrix}, \tag{16}$$

where $I_{tk}$ is the $(tk \times tk)$ identity matrix, 0 is an appropriately-sized matrix of 0s, and $R$ is some appropriately-sized binary matrix. The rows of $I_{tk}$ represent that adversaries know each other's values, and $R$ represents the edges between adversaries.

**Theorem 2.** *Let $i < tn$. Then $\theta_i$ has a solution in $A$ if and only if $\theta_i$ has a solution in $A'$.*

Proof. Observe that

$$\text{rref}(A') = \begin{bmatrix} \text{rref}(A) & 0 \\ 0 & I_{tk} \end{bmatrix}, \tag{17}$$

ignoring row-switching transformations. The bottom $tk$ rows solve exactly $\theta_i$ in $A$ for $i \geq tn$. The upper rows solve $\theta_i$ in $A$ for $i < tn$ if and only if the rows of $\text{rref}(A)$ do so. □

Intuitively, Theorem 2 holds because the linear dependencies that exist within $A$ remain unaffected by $R$.

### 4.4 Reconstruction Attack Feasibility

We show that reconstruction is feasible for honest-but-curious adversaries. We run the attack in static graphs with randomly-placed adversaries passively collecting data. We measure both the success rate and the number of rounds until success. Our source code is publicly available [22].

*Remark* 3. This section pertains only to static graphs. We show a reduction from edge-dynamic graphs to static graphs in Section 5.4.

*4.4.1 Experimental setup.* By Theorem 1, the success rate of the attack depends only on the adversaries' direct neighbourhood. Therefore, instead of modeling large peer-to-peer networks, it suffices to model only the subgraph that is relevant for the attack. Additionally, by Theorem 2, edges between adversaries can be ignored. Therefore, given any graph $G = (V, E)$ and a set of colluding adversaries $C \subseteq V$, it suffices to model the induced subgraph $G[C]$, minus edges between adversaries. This forms a bipartite graph $H$. We provide an example of graph induction in Figure 3.

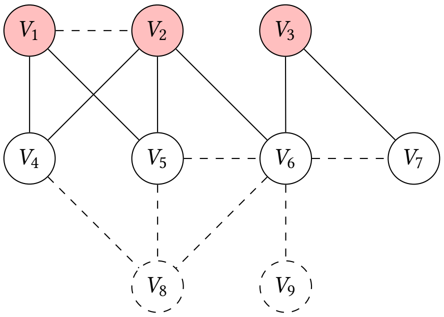


**Figure 3: A graph $G$. Adversaries $C = \{V_1, V_2, V_3\}$ are shaded. The bipartite subgraph $H = G[C]$ consists of exactly the non-dotted nodes and edges.**

We emphasise that reconstruction depends only on the adversaries' view, regardless of the remaining graph outside this view. However, the likelihood of obtaining any specific adversarial view *does* depend on the full graph. For example, the probability that a random adversarial view contains a cycle depends on the connectivity of the full graph. For our experiments, we choose not to make assumptions on the graph's topology, analysing all possible adversarial views equally, so that our results are agnostic to the specific network, application, and adversary.

Bipartite graphs can be parameterised by three variables: the number of adversaries, the number of direct neighbours, and the number of edges. We generate random graphs according to these parameter, subject to some filtering:

- We exclude graphs in which there is a adversary with only one edge because this would allow trivial attacks, as described in Section 3.3.
- We do *not* exclude graphs in which there is an honest-but-curious user with only one edge, because this user may have more edges in $G$ that are not in $H$.
- We exclude graphs in which an honest-but-curious user has no neighbours, because these cases do not accurately represent the bipartite graph's parameters.
- We do *not* exclude graphs in which an adversary has no neighbours.
- We do *not* exclude disconnected graphs.

*4.4.2 Amount of reconstructed data.* For our first experiment, we measure the amount of private data that adversaries can reconstruct. We generate a large amount of random bipartite graphs as described above, and count the number of partial solutions in the biadjacency matrices. This corresponds to the adversarial knowledge if neighbours do not update their values, and thus represents the strongest reconstruction attack that adversaries can perform. In Section 4.4.3 we also consider neighbours updating their values.

Firstly, we look at the proportion of data that can be reconstructed, shown in Figure 4. We see that if the number of adversaries is close to the number of neighbours, the adversary is typically able to reconstruct all neighbours' data. As the number of neighbours increases, fewer data can be reconstructed, unless compensated for by a higher connectivity. If the graph has many neighbours and few edges, adversaries share fewer neighbours, and are thus typically unable to exploit the overlaps in their aggregates.

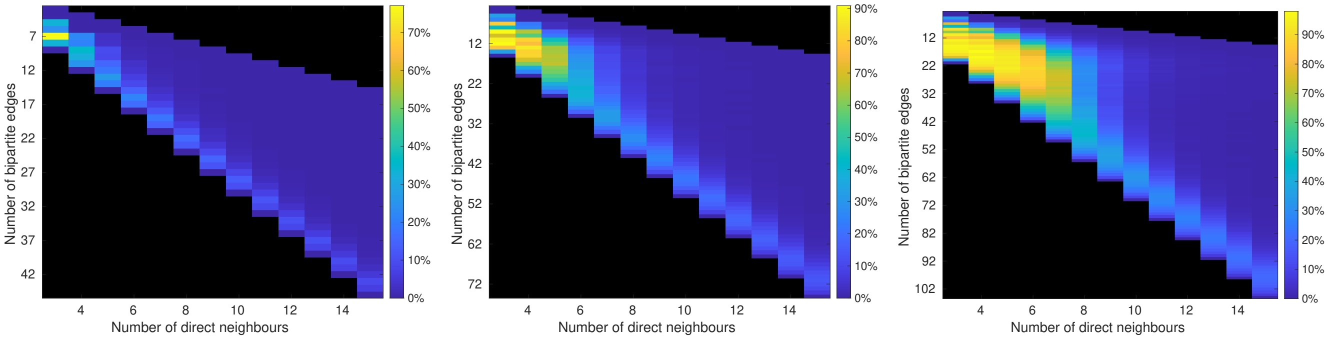
Secondly, we look at the distribution of how much data can be reconstructed, shown in Figure 5. We see again that adversaries are more successful if they outnumber their neighbours. As the number of neighbours increases, so does the probability of being unable to reconstruct any data. However, even if three adversaries passively observe 15 neighbours, they still have an 11.0% probability of reconstructing at least one neighbour's datum, which is unacceptable for any privacy-preserving scheme.

*4.4.3 Rounds until first reconstruction.* Some partial solutions are harder to obtain than others. For example, if the graph is such that users update their values faster than adversaries can collect them, adversaries may never "converge" to a (partial) solution.

In the next experiment, we measure how many rounds adversaries need before reconstruction succeeds. For each of the subgraphs in Figure 4 that were found to be susceptible to the attack, we simulate a multi-party summation protocol as follows. Each round, a uniformly random user in the subgraph wakes up. If the user is an adversary, they learn the sum of their neighbours' values, and adds this to the adversarial knowledge. Otherwise, if a non-adversary wakes up, we simulate an update: The next adversarial sum that includes this non-adversary will use a new column in the adversarial knowledge matrix. After every round, the adversaries check for a partial solution. We repeat this procedure 100 times to control for the order in which users wake up, truncate instances that have no partial solutions after 250 rounds, and take the mean number of rounds until the first partial solution is found.

We show the mean number of rounds until the reconstruction attack succeeds in Figure 6. We see that the attack is fastest when there are more adversaries, more edges, and fewer neighbours. Intuitively, this means that the required number of summations increases if neighbours can update their values at a higher rate than adversaries can observe them. For example, 3 adversaries against 15 neighbours require on average 8.8 rounds before they can reconstruct private data. In related works such as [11, 16, 48], users run hundreds or thousands of rounds before the protocol terminates, significantly more than required in our attack.

*4.4.4 Conclusion of results.* We sampled all possible views of randomly selected adversaries in random graphs, excluding some trivial attack cases. If the reconstruction attack succeeds, the adversaries obtain other users' private inputs to the information-theoretically secure summation operation. Our results show that passive honest-but-curious adversaries are able to obtain private data in this scenario with non-negligible probability. While we note that different classes of graph topologies may have varying susceptibility to reconstruction attacks, we conclude that, in general, individually protecting each summation is insufficient for confidentiality.

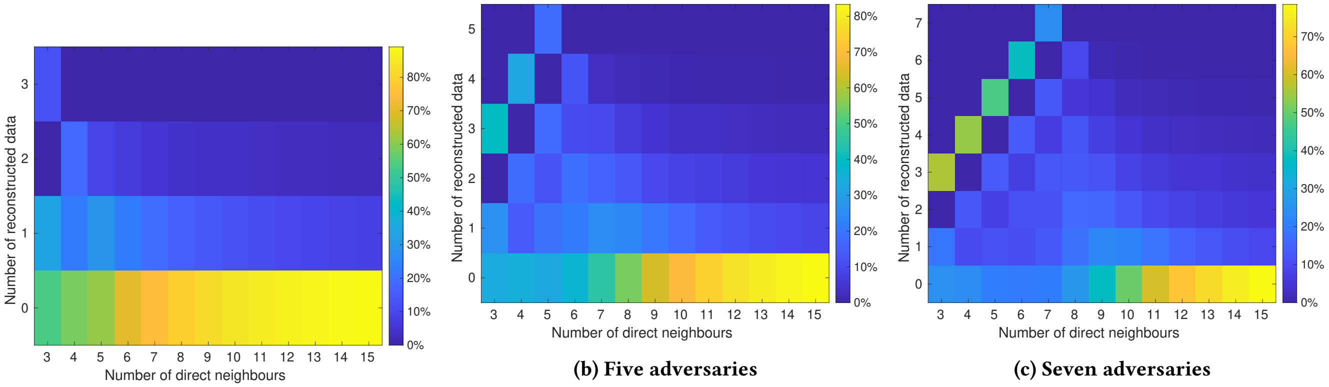


(a) Three adversaries

(b) Five adversaries

(c) Seven adversaries

**Figure 4: Proportion of neighbours' private data that can be reconstructed by adversaries. Each point represents the mean over 1000 random bipartite graphs. Black points indicate no valid bipartite graphs could be found. Note the different y-axes.**

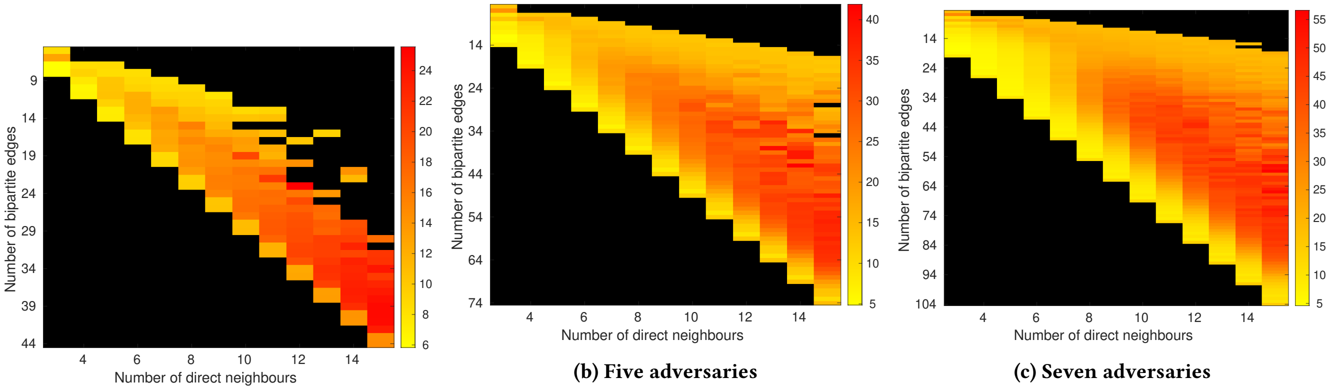


(a) Three adversaries

(b) Five adversaries

(c) Seven adversaries

**Figure 5: Probability of reconstructing a given number of neighbours' data, ignoring the number of edges. Each column adds up to 100%, and corresponds to a column in Figure 4.**



(a) Three adversaries

(b) Five adversaries

(c) Seven adversaries

**Figure 6: Mean number of adversarial summations needed to obtain private data. Each point corresponds to 100 attacks on each of the solvable graphs from Figure 4.**

# 5 Girth as a Peer-to-Peer Reconstruction Countermeasure

In a centralised protocol, the single aggregator can track which summations have occurred, and refuse a subsequent summation if it would result in a partial solution. However, in a distributed computation, there is no such aggregator, and simulating the aggregator using a multi-party protocol is impractical as this would require involving all users in each summation. In this section, we show that to prevent reconstruction it is sufficient to increase the network's girth, which is the length of the network's shortest cycle. The network's girth is an established metric for peer-to-peer networks, with various peer-to-peer algorithms for measuring and increasing the girth [9, 24, 36, 41]. Using such an algorithm before running a privacy-preserving dynamic-data multi-party summation protocol is thus sufficient to prevent reconstruction of private data by honest-but-curious adversaries.

We begin in Section 5.1 by showing that reconstruction requires collusion. In Section 5.2, we show that reconstruction does not work in acyclic graphs, regardless of the number adversaries. In Section 5.3, generalise results to determine an upper bound on the number of adversaries. In Section 5.4, consider graphs with dynamic edges. Finally, in Section 5.5, we briefly evaluate the impact that increasing girth has on distributed convergence.

## 5.1 Privacy in Static Graphs without Collusion

We begin by considering the special case of $k = 1$, i.e. a setting without collusion. We show that, if the graph is static, the adversary cannot obtain other users' private values regardless of topology, barring trivial attacks.

Assuming a privacy-preserving summation protocol, it is self-evident that repeating the summation over the same set of values does not leak any private data. However, while the set of neighbours is always the same in the static no-collusion setting, neighbours still update their local values. Thus, it remains to be shown that no reconstruction is possible with this kind of composition.

**Lemma 1.** Given adversarial knowledge $A \in \mathbb{R}^{t \times nt}$ of a single adversary with $n \geq 2$ fixed neighbours, we have for any $y \in \mathbb{R}^{1 \times t}$

$$\forall \mu, \nu \in [\![n]\!] : \sum_{i \in [\![t]\!]} (yA)_{\mu t+i} = \sum_{i \in [\![t]\!]} (yA)_{\nu t+i}. \tag{18}$$

Here, $\sum_{i \in [\![t]\!]} (yA)_{\nu t+i}$ is the sum of components of $yA$ relating to neighbour $\nu$. The equation states that in any linear combination $yA$, every neighbour has the same sum of components.

Proof. Firstly, because the adversary has fixed neighbours,

$$\forall \tau \in [\![t]\!], \nu \in [\![n]\!] : \sum_{i \in [\![t]\!]} A_{\tau, \nu t+i} = 1. \tag{19}$$

In the linear combination $yA$, the rows of $A$ are scaled according to $y$ and then summed together. Therefore, since each row includes each neighbour exactly once,

$$\forall \nu \in [\![n]\!] : \sum_{i \in [\![t]\!]} (yA)_{\nu t+i} = \sum_{\tau \in [\![t]\!]} y_\tau. \tag{20}$$

□

**Corollary 1.** Given adversarial knowledge $A \in \mathbb{R}^{t \times nt}$ of a single adversary with $n \geq 2$ fixed neighbours, there exists no $y \in \mathbb{R}^{1 \times t}$ such that $yA$ has exactly one non-zero value. Therefore, there exist no partial solutions for $A$.

## 5.2 Privacy in Static Graphs with Unbounded Collusion

The special case of $k = 1$ provides some insights into the workings of the reconstruction attack, but not allowing any collusion is not realistic, as honest-but-curious collusion in the form of secretly exchanging information is undetectable and there are no strong incentives against it. Therefore, we now proceed to consider the general case of $k \geq 1$.

Partial solutions are linear combinations of the rows of the adversarial knowledge such that all but one column cancels out, as in Equation 1. We already know from Corollary 1 that a partial solution requires multiple adversaries. If two rows in the adversarial knowledge from different adversaries match in multiple columns, then these adversaries share multiple neighbours, and the graph has a cycle. Otherwise, if no two rows from different adversaries overlap in multiple columns, then, since each equation has at least two non-zero columns, each equation introduces new unknowns, taking the adversaries further from a partial solution. In this case, if the adversaries are able to find a partial solution, they must have another row that cancels out the unknowns of multiple other rows; but this, too, introduces a cycle. The intuition thus seems to be that partial solutions require a cyclic graph. We now formally prove that this intuition is correct.

**Theorem 3.** Let $G = (V_G, E_G)$ be an undirected graph, let $C \subseteq V_G$ be the set of adversaries, let $k := |C|$, let $n := |N_G(C)|$, let $t$ be the number of summations performed by the adversaries $C$, and let $A \in \mathbb{R}^{t \times nt}$ be the adversarial knowledge.

If $G$ is acyclic, then $A$ does not have partial solutions.

Proof. We give a proof by contraposition: Given a partial solution to $A$, we show that $G$ is cyclic. Let $y \in \mathbb{R}^{1 \times t}$ be a partial solution to $A$. We show how to find a bipartite subgraph $H$ of $G$ such that its biadjacency matrix $A''$ has a partial solution $y''$. We then show that this implies the existence of a cycle in $G$. Our proof works in multiple steps: (1) combine columns of $A$ to create $A'$, (2) remove rows from $A'$ to create $A''$, (3) create the corresponding partial solution $y''$, and finally (4) show that $G$ is cyclic. We show an example of this procedure in Figure 7.

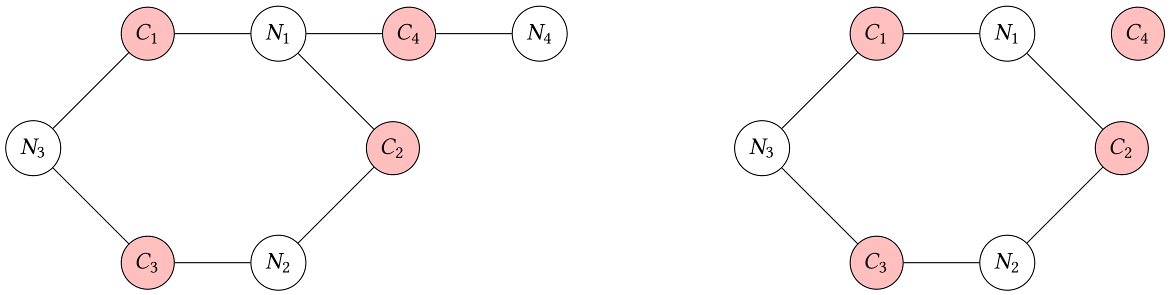
(1) *Combine columns.* We merge the $t$ columns in $A$ assigned to each neighbour to obtain $A'$. Let $y' = y$, and let $A' \in \mathbb{R}^{t \times n}$ such that

$$\forall \tau \in [\![t]\!], \nu \in [\![n]\!] : A'_{\tau,\nu} := \sum_{i \in [\![t]\!]} A_{\tau, \nu t+i}. \tag{21}$$

It follows from Property 1 that this is a binary matrix, and it follows from Property 2 that no neighbour relations are removed. Furthermore, observe that

$$\forall \nu \in [\![n]\!] : (y'A')_\nu = \sum_{i \in [\![t]\!]} (yA)_{\nu t+i}. \tag{22}$$

Since $yA$ contains exactly one non-zero value, so does $y'A'$. Therefore, $y'$ is a partial solution to $A'$.



**(a) A graph $G$ featuring adversaries $C = \{C_1, C_2, C_3, C_4\}$ and non-adversaries $N = \{N_1, N_2, N_3, N_4\}$.**

**(d) The bipartite graph $H$ corresponding to biadjacency matrix $A''$.**

$$A = \left[\begin{array}{ccccc|ccccc|ccccc|ccccc}
\multicolumn{5}{c}{\overbrace{\hphantom{0\ 0\ 0\ 0\ 0}}^{N_1}} & \multicolumn{5}{c}{\overbrace{\hphantom{0\ 0\ 0\ 0\ 0}}^{N_2}} & \multicolumn{5}{c}{\overbrace{\hphantom{0\ 0\ 0\ 0\ 0}}^{N_3}} & \multicolumn{5}{c}{\overbrace{\hphantom{0\ 0\ 0\ 0\ 0}}^{N_4}} \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}\right], \quad
A' = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad
A'' = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

**(b) The adversarial knowledge $A$ after users from Figure 7a run in the sequence $(C_1, C_2, C_3, N_3, C_3, C_4)$; the matrix $A'$ with collapsed columns; and the matrix $A''$ without duplicate and unused rows.**

$$y = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 \end{bmatrix}, \qquad y' = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 \end{bmatrix}, \qquad y'' = \begin{bmatrix} 1 & 1 & -1 \end{bmatrix}$$

**(c) Partial solutions respectively of $A$, $A'$, and $A''$.**

**Figure 7: Example transformation of graph and adversarial knowledge as seen in the proof of Theorem 3.**

(2) *Remove rows.* We remove duplicate and unused rows from $A'$ to obtain $A''$. We define $A''$ as a set of rows:

$$\begin{aligned}
A'' \coloneqq \{A'_i \mid i \in \llbracket t \rrbracket \land & \quad (23) \\
\nexists j \in \llbracket i \rrbracket : A'_i = A'_j \land & \quad (24) \\
\textstyle\sum \{y'_j \mid j \in \llbracket t \rrbracket \land A'_i = A'_j\} \neq 0\}. & \quad (25)
\end{aligned}$$

Here, Equation 24 excludes duplicates by only choosing row $A'_i$ if there is no $j < i$ such that $A'_i = A'_j$, and Equation 25 excludes unused rows by only picking row $A'_i$ if the sum of $y'_j$ over all identical rows $A'_j$ is non-zero.

(3) *Create partial solution.* We similarly combine and remove the corresponding columns from $y'$ to obtain $y''$. To do so, we define a function $\phi$ that describes how the rows of $A''$ relate to the rows of $A'$. Let $s$ be the number of rows in $A''$. Then we define $\phi : \llbracket s \rrbracket \to \llbracket t \rrbracket^*$ such that

$$\forall \tau \in \llbracket t \rrbracket, \sigma \in \llbracket s \rrbracket : \tau \in \phi(\sigma) \Leftrightarrow A'_\tau = A''_\sigma. \tag{26}$$

Using this function, we define $y'' \in \mathbb{R}^{1 \times s}$ as

$$\forall \sigma \in \llbracket s \rrbracket : y''_\sigma \coloneqq \sum_{\tau \in \phi(\sigma)} y'_\tau. \tag{27}$$

It follows that

$$\begin{aligned}
\forall v \in \llbracket n \rrbracket : (y''A'')_v &= \sum_{\sigma \in \llbracket s \rrbracket} (y''_\sigma A''_{\sigma,v}) && (28) \\
&= \sum_{\sigma \in \llbracket s \rrbracket} \sum_{\tau \in \phi(\sigma)} (y'_\tau A''_{\sigma,v}) && (29) \\
&= \sum_{\sigma \in \llbracket s \rrbracket} \sum_{\tau \in \phi(\sigma)} (y'_\tau A'_{\tau,v}) && (30) \\
&= \sum_{\tau \in \llbracket t \rrbracket} (y'_\tau A'_{\tau,v}) && (31) \\
&= (y'A')_v. && (32)
\end{aligned}$$

Therefore, $y''A'' = y'A'$, and $y''$ is a partial solution to $A''$.

(4) *Find cycle.* Note that $A''$ is the biadjacency matrix of some bipartite subgraph $H = (C', N_G(C), E_H)$ of $G$, where $C' \subseteq C$ and $E_H \subseteq E_G$. Assume, for the sake of contradiction, that $H$ is acyclic. Then $H$ has two distinct nodes $i, j$ with degree one. Since adversaries cannot have degree one in $G$, and $\forall c \in C' : (N_H(c) = N_G(c) \lor N_H(c) = \emptyset)$, we know that $i, j \in N_G(C)$. Consequently, the columns in $A''$ for $i, j$ must each contain only one non-zero value, and $y''$ does not contain zeroes at all by Equation 25. Therefore, $(y''A'')_i \neq 0$ and $(y''A'')_j \neq 0$. However, this implies that $y''A''$ has multiple non-zero values, which contradicts the earlier observation that $y''$ is a partial solution to $A''$. Therefore, $H$ is cyclic, and so is $G$.

□

Our proof shows that partial solutions imply the existence of cycles. However, this does not mean that cycles imply the existence of partial solutions. Indeed, we show in Section 5.3 that structured cycles can be introduced without creating partial solutions.

*Remark* 4. Theorem 3 pertains only to *partial* solutions. Even in an acyclic topology, there may be linear relations that reveal sensitive information without leaking private values outright, such as $\theta_1 = \theta_2$ or $\theta_3 = 4 \times \theta_5$. Protecting these relations is left for future work.

## 5.3 Privacy in Static Graphs with Bounded Collusion

While acyclic graphs resist reconstruction attacks, these graphs are not well-suited for peer-to-peer networks for two reasons. Firstly, if any non-leaf node becomes unavailable, the network becomes disconnected. Secondly, leaf nodes have only one neighbour, and thus cannot initiate summations to learn from their neighbours.

We show that no partial solutions exist given an upper bound on the number of adversaries. This bound depends on the graph's girth, which is the length of its shortest cycle.

**Theorem 4.** Let $G = (V_G, E_G)$ be an undirected graph, let $C \subseteq V_G$ be a set of $k$ adversaries, let $n := |N_G(C)|$, let $t$ be the number of summations performed by $C$, and let $A \in \mathbb{R}^{t \times nt}$ be the adversarial knowledge.

If $\text{girth}(G) > 2k$, then $A$ does not have partial solutions.

Proof. We give a proof by contraposition: Given a partial solution to $A$, we show that $\text{girth}(G) \leq 2k$. Let $H$ be as in the proof of Theorem 3. Then $H$ is cyclic. Since $H$ is bipartite, every edge in the cycle is between an adversary and a neighbour. Since each node in the cycle is visited at most once, the cycle length is at most $2k$. This cycle also exists in $G$. Therefore, $\text{girth}(G) \leq 2k$. □

## 5.4 Privacy in Dynamic Graphs

So far, we have assumed that graphs are static. However, this prevents users from changing their neighbours, which is unrealistic if users move through the network. We briefly show that dynamic graphs can be reduced to static graphs.

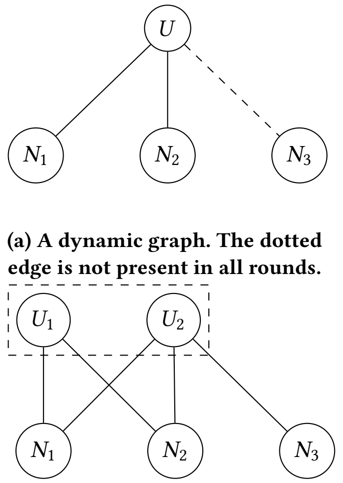
If a single user performs two summations over two sets of neighbours, they learn exactly the same information as two users would over those same sets of neighbours. We show an example in Figure 8. More generally, $k$ users with static neighbours can learn the exact same information as $\ell$ users with $k$ different sets of neighbours. Our results on reconstruction feasibility in static graphs from Section 4.4 can be translated similarly to dynamic graphs.

We conclude that Theorem 4 implies the following.

**Corollary 2.** Let $G = (V_G, E_G)$ be a dynamic undirected graph, let $C \subseteq V_G$ be a set of adversaries, let $n := |N_G(C)|$, let $t$ be the number of summations performed by $C$, let $k$ be the number of sets of neighbours the adversaries sum over, and let $A \in \mathbb{R}^{t \times nt}$ be the adversarial knowledge.

If $\text{girth}(G) > 2k$, then $A$ does not have partial solutions.

There are several important limitations to this result. Firstly, the upper bound on the number of adversaries depends on the girth, but the girth may not be known beforehand if users move through the network in unpredictable ways. Secondly, even if a minimum girth is guaranteed throughout the protocol, the upper bound implies a maximum number of changes that may occur during the protocol.



**(a) A dynamic graph. The dotted edge is not present in all rounds.**

**(b) A reduction to a static graph. $U$ has been split into $U_1$ and $U_2$.**

**Figure 8: Example of how a dynamic graph can be reduced to a static graph. $U$ learns the same as $U_1$ and $U_2$ together.**

## 5.5 Impact on Convergence

We briefly evaluate the impact of increasing the network's girth on the convergence of a protocol running over that network. Specifically, we numerically simulate a distributed averaging protocol [53], which is just a non-privacy-preserving form of distributed learning. We intentionally choose a simple, efficient, non-noisy protocol to make the impact of the girth parameter most apparent. The "numerical simulation" part of the description is because we do not actually create separate processes and communication for the nodes. Our source code is publicly available [22].

We use the system model presented in Section 3.3. We create a network by generating a random Erdős–Rényi graph with 50 nodes and with each edge having a probability $p$ of being added. Each node holds a single private scalar value, sampled uniformly from the range $\{0 \ldots 50\}$. Each round, one random node updates their private value to be the unweighted mean of their neighbours' values and their own value. We then measure the number of rounds until convergence, and take the mean over 1000 repetitions of this procedure. We define convergence as the moment at which any two nodes' local values differ by at most 1. Changing this threshold does not give fundamentally different results.

To measure the effect girth has on convergence, we "stretch" graphs to a given girth by iteratively removing random edges from cycles shorter than the desired girth until no such cycles remain. With 50 nodes, stretching to a girth of $x$ ensures reconstruction attacks are impossible when less than ${}^{x/2}/_{50} = x\%$ of users collude. For example, after stretching the girth to 25, the graph can resist collusions of less than 25% of users.

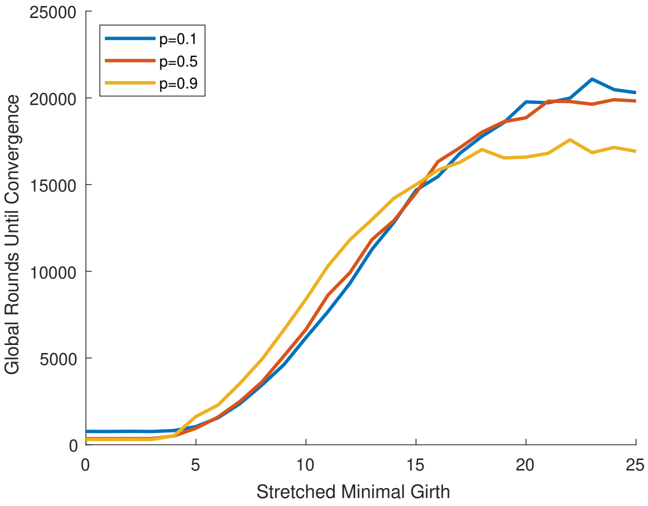


**Figure 9: Number of rounds until convergence in distributed averaging in random Erdős–Rényi graphs with 50 nodes and varying edge probabilities $p$, as a function of the girth to which the graphs are "stretched".**

We show our results in Figure 9. Since undirected graphs always have girth at least 3, no significant changes occur at these low girths. As the girth increases, so does the number of rounds required. As the girth approaches 25, the slope approaches zero. Graphs that initially have more edges (as determined by $p$) require more rounds at low girths, but settle at a lower number of rounds at high girths. When we look at our experiments in more detail, we see that ceilings occur once all cycles have been removed, and that graphs with high $p$ retain more edges. This matches the intuition that information propagates more efficiently when there are more edges.

Our results show that increasing girth affects convergence speed significantly. Though state-of-the-art distributed learning protocols typically require several tens of thousands of rounds [11, 16, 48], the magnitude by which increasing girth increases the number of required rounds may be excessive for some applications. More sophisticated edge removal methods may ameliorate this issue. Furthermore, though implementing the cycle removal method from our experiment above as a distributed protocol is trivial,[2] these methods are not necessarily communicationally efficient. To the best of our knowledge, there is no research on communication-efficient distributed "graph stretching". That said, there are distributed protocols for measuring the network's girth [9] and for removing *all* cycles [24, 41]. We conclude that determining a network's resistance by measuring the girth is feasible in general, but increasing girth is practical only when communication efficiency is not a concern.

## 6 Conclusion

We investigated reconstruction attacks in the setting of secure multi-party computation. We observed that existing multi-party computation literature does not consider protocols in which intermediate values are intentionally exposed by the ideal functionality, and seemingly assumes that protocols are not self-composed when deployed. In our investigation, we focused on a peer-to-peer setting with privacy-preserving summation in which users' data change over time. In random subgraphs with 18 users, we found that three passive honest-but-curious adversarial users have an 11.0% success rate at recovering another user's private data using a reconstruction attack, requiring an average of 8.8 rounds per adversary. We analysed the structural dependencies of the underlying network graph that permit this attack, and proved that successful reconstruction attacks correspond to cycles in the network. More generally, we showed that the length of the graph's shortest cycle determines the minimum number of adversaries required for the attack. We conclude that removing short cycles from the network is a feasible countermeasure, albeit with considerable cost towards the convergence speed of distributed protocols.

Our work sets the first step towards preventing reconstruction in the peer-to-peer setting as seen in multi-party computation, and opens up multiple questions for future work. Firstly, and most obviously, though we have found a sufficient criterion to determine reconstruction feasibility, finding a criterion that is also necessary would allow using some graphs which our criterion currently forbids. Secondly, our work is limited to a strictly syntactic notion of privacy, and does not protect linear relations between data, which is required to protect against adaptive adversaries. Thirdly, though our restriction to the summation operation is already sufficient to analyse decentralised learning, our work could be extended to cover compositions with other operations, such as multiplication or comparison. Finally, the addition of differentially private noise may further strengthen the provided level of privacy.

## Acknowledgments

This paper is partly supported by the European Union's Horizon Europe research and innovation program under grant agreement No. 101094901, the SEPTON project (Security Protection Tools for Networked Medical Devices).

## References

[1] Michael Backes, Birgit Pfitzmann, and Michael Waidner. 2007. The reactive simulatability (RSIM) framework for asynchronous systems. *Inf. Comput.* 205, 12 (2007), 1685–1720. https://doi.org/10.1016/j.ic.2007.05.002

[2] James Henry Bell, Kallista A. Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. 2020. Secure Single-Server Aggregation with (Poly)Logarithmic Overhead. In *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna (Eds.). ACM, 1253–1269. https://doi.org/10.1145/3372297.3417885

[3] Aurélien Bellet, Rachid Guerraoui, Mahsa Taziki, and Marc Tommasi. 2018. Personalized and Private Peer-to-Peer Machine Learning. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain (Proceedings of Machine Learning Research, Vol. 84)*, Amos J. Storkey and Fernando Pérez-Cruz (Eds.). PMLR, 473–481. http://proceedings.mlr.press/v84/bellet18a.html

[4] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. 2005. Practical privacy: the SuLQ framework. In *Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 13-15, 2005, Baltimore, Maryland, USA*, Chen Li (Ed.). ACM, 128–138. https://doi.org/10.1145/

[2] A node can break all cycles of at most length $\ell$ that they are part of as follows. The node floods a unique random message, paired with a counter starting at $\ell$, through the network. Each time a node forwards the message, the counter is decreased. Once the counter reaches zero, nodes stop forwarding the message. If (and only if) the source node receives back their own message, they are part of a cycle of length at most $\ell$, and remove the edge on which the message came in.

1065167.1065184

[5] Dan Bogdanov, Peeter Laud, Sven Laur, and Pille Pullonen. 2014. From Input Private to Universally Composable Secure Multi-party Computation Primitives. In *IEEE 27th Computer Security Foundations Symposium, CSF 2014, Vienna, Austria, 19-22 July, 2014*. IEEE Computer Society, 184–198. https://doi.org/10.1109/CSF.2014.21

[6] Kallista A. Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy Preserving Machine Learning. *IACR Cryptol. ePrint Arch.* (2017), 281. http://eprint.iacr.org/2017/281

[7] Stephen P. Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. 2006. Randomized gossip algorithms. *IEEE Trans. Inf. Theory* 52, 6 (2006), 2508–2530. https://doi.org/10.1109/TIT.2006.874516

[8] Ran Canetti. 2001. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*. IEEE Computer Society, 136–145. https://doi.org/10.1109/SFCS.2001.959888

[9] Keren Censor-Hillel, Orr Fischer, Tzlil Gonen, François Le Gall, Dean Leitersdorf, and Rotem Oshman. 2021. Fast Distributed Algorithms for Girth, Cycles and Small Subgraphs. *CoRR* abs/2101.07590 (2021). arXiv:2101.07590 https://arxiv.org/abs/2101.07590

[10] Xuhui Chen, Jinlong Ji, Changqing Luo, Weixian Liao, and Pan Li. 2018. When Machine Learning Meets Blockchain: A Decentralized, Privacy-preserving and Secure Design. In *IEEE International Conference on Big Data (IEEE BigData 2018), Seattle, WA, USA, December 10-13, 2018*, Naoki Abe, Huan Liu, Calton Pu, Xiaohua Hu, Nesreen K. Ahmed, Mu Qiao, Yang Song, Donald Kossmann, Bing Liu, Kisung Lee, Jiliang Tang, Jingrui He, and Jeffrey S. Saltz (Eds.). IEEE, 1178–1187. https://doi.org/10.1109/BigData.2018.8622598

[11] Hsin-Pai Cheng, Patrick Yu, Haojing Hu, Feng Yan, Shiyu Li, Hai Li, and Yiran Chen. 2018. LEASGD: an Efficient and Privacy-Preserving Decentralized Algorithm for Distributed Learning. *CoRR* abs/1811.11124 (2018). arXiv:1811.11124 http://arxiv.org/abs/1811.11124

[12] Francis Y. L. Chin. 1978. Security in Statistical Databases for Queries with Small Counts. *ACM Trans. Database Syst.* 3, 1 (1978), 92–104. https://doi.org/10.1145/320241.320250

[13] Francis Y. L. Chin and Gultekin Özsoyoglu. 1982. Auditing and Inference Control in Statistical Databases. *IEEE Trans. Software Eng.* 8, 6 (1982), 574–582. https://doi.org/10.1109/TSE.1982.236161

[14] Chris Clifton and Tamir Tassa. 2013. On Syntactic Anonymity and Differential Privacy. *Trans. Data Priv.* 6, 2 (2013), 161–183. http://www.tdp.cat/issues11/abs.a124a13.php

[15] Graham Cormode, Cecilia M. Procopiuc, Entong Shen, Divesh Srivastava, and Ting Yu. 2013. Empirical privacy and empirical utility of anonymized data. In *Workshops Proceedings of the 29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013*, Chee Yong Chan, Jiaheng Lu, Kjetil Nørvåg, and Egemen Tanin (Eds.). IEEE Computer Society, 77–82. https://doi.org/10.1109/ICDEW.2013.6547431

[16] Edwige Cyffers, Aurélien Bellet, and Jalaj Upadhyay. 2024. Differentially Private Decentralized Learning with Random Walks. *CoRR* abs/2402.07471 (2024). arXiv:2402.07471 https://arxiv.org/abs/2402.07471

[17] Edwige Cyffers, Mathieu Even, Aurélien Bellet, and Laurent Massoulié. 2022. Muffliato: Peer-to-Peer Privacy Amplification for Decentralized Optimization and Averaging. In *NeurIPS*. https://proceedings.neurips.cc/paper/2022/hash/65d32185f73cbf4535449a792c63926f-Abstract-Conference.html

[18] Josenildo Costa da Silva, Matthias Klusch, Stefano Lodi, and Gianluca Moro. 2004. Inference Attacks in Peer-to-Peer Homogeneous Distributed Data Mining. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004*, Ramón López de Mántaras and Lorenza Saitta (Eds.). IOS Press, 450–454.

[19] George Danezis. 2003. Statistical Disclosure Attacks: Traffic Confirmation in Open Environments. In *Security and Privacy in the Age of Uncertainty, IFIP TC11 $18^{th}$ International Conference on Information Security (SEC2003), May 26-28, 2003, Athens, Greece (IFIP Conference Proceedings, Vol. 250)*, Dimitris Gritzalis, Sabrina De Capitani di Vimercati, Pierangela Samarati, and Sokratis K. Katsikas (Eds.). Kluwer, 421–426.

[20] Gábor Danner, Árpád Berta, István Hegedüs, and Márk Jelasity. 2018. Robust Fully Distributed Minibatch Gradient Descent with Privacy Preservation. *Secur. Commun. Networks* 2018 (2018), 6728020:1–6728020:15. https://doi.org/10.1155/2018/6728020

[21] Florine W. Dekker and Zekeriya Erkin. 2021. Privacy-Preserving Data Aggregation with Probabilistic Range Validation. In *Applied Cryptography and Network Security - 19th International Conference, ACNS 2021, Kamakura, Japan, June 21-24, 2021, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12727)*, Kazue Sako and Nils Ole Tippenhauer (Eds.). Springer, 79–98. https://doi.org/10.1007/978-3-030-78375-4_4

[22] Florine W. Dekker, Zekeriya Erkin, and Mauro Conti. 2024. *Source code underlying the publication: Topology-Based Reconstruction Prevention for Decentralised Learning*. https://doi.org/10.4121/21572601.v2

[23] Dorothy E. Denning. 1980. Secure Statistical Databases with Random Sample Queries. *ACM Trans. Database Syst.* 5, 3 (1980), 291–315. https://doi.org/10.1145/320613.320616

[24] Shlomi Dolev and Ronen I. Kat. 2008. HyperTree for self-stabilizing peer-to-peer systems. *Distributed Comput.* 20, 5 (2008), 375–388. https://doi.org/10.1007/s00446-007-0038-9

[25] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 4052)*, Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener (Eds.). Springer, 1–12. https://doi.org/10.1007/11787006_1

[26] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* 9, 3-4 (2014), 211–407. https://doi.org/10.1561/0400000042

[27] Alexandre V. Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. 2003. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*, Frank Neven, Catriel Beeri, and Tova Milo (Eds.). ACM, 211–222. https://doi.org/10.1145/773153.773174

[28] Ivan P. Fellegi. 1972. On the question of statistical confidentiality. *J. Amer. Statist. Assoc.* 67, 337 (1972), 7–18. https://doi.org/10.1080/01621459.1972.10481199

[29] Flavio D. Garcia and Bart Jacobs. 2010. Privacy-Friendly Energy-Metering via Homomorphic Encryption. In *Security and Trust Management - 6th International Workshop, STM 2010, Athens, Greece, September 23-24, 2010, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 6710)*, Jorge Cuéllar, Gilles Barthe, and Alexander Pretschner (Eds.). Springer, 226–238. https://doi.org/10.1007/978-3-642-22444-7_15

[30] Shangwei Guo, Tianwei Zhang, Guowen Xu, Han Yu, Tao Xiang, and Yang Liu. 2022. Topology-Aware Differential Privacy for Decentralized Image Classification. *IEEE Trans. Circuits Syst. Video Technol.* 32, 6 (2022), 4016–4027. https://doi.org/10.1109/TCSVT.2021.3105723

[31] Briland Hitaj, Giuseppe Ateniese, and Fernando Pérez-Cruz. 2017. 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*, Bhavani Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu (Eds.). ACM, 603–618. https://doi.org/10.1145/3133956.3134012

[32] Adel Jebali, Salma Sassi, and Abderrazak Jemai. 2019. Inference Control in Distributed Environment: A Comparison Study. In *Risks and Security of Internet and Systems, 14th International Conference, CRiSIS 2019, Hammamet, Tunisia, October 29-31, 2019, Proceedings (Lecture Notes in Computer Science, Vol. 12026)*, Slim Kallel, Frédéric Cuppens, Nora Cuppens-Boulahia, and Ahmed Hadj Kacem (Eds.). Springer, 69–83. https://doi.org/10.1007/978-3-030-41568-6_5

[33] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. 2021. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* 14, 1-2 (2021), 1–210. https://doi.org/10.1561/2200000083

[34] Renuga Kanagavelu, Zengxiang Li, Juniarto Samsudin, Yechao Yang, Feng Yang, Rick Siow Mong Goh, Mervyn Cheah, Praewpiraya Wiwatphonthana, Khajonpong Akkarajitsakul, and Shangguang Wang. 2020. Two-Phase Multi-Party Computation Enabled Privacy-Preserving Federated Learning. In *20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing, CCGRID 2020, Melbourne, Australia, May 11-14, 2020*. IEEE, 410–419. https://doi.org/10.1109/CCGrid49817.2020.00-52

[35] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. 2008. What Can We Learn Privately?. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*. IEEE Computer Society, 531–540. https://doi.org/10.1109/FOCS.2008.27

[36] Felix Lazebnik and Vasiliy A. Ustimenko. 1995. Explicit Construction of Graphs with an Arbitrary Large Girth and of Large Size. *Discret. Appl. Math.* 60, 1-3 (1995), 275–284. https://doi.org/10.1016/0166-218X(94)00058-L

[37] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. 2017. Can Decentralized Algorithms Outperform Centralized Algorithms? A Case

Study for Decentralized Parallel Stochastic Gradient Descent. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5330–5340. https://proceedings.neurips.cc/paper/2017/hash/f75526659f31040afeb61cb7133e4e6d-Abstract.html

[38] Yehuda Lindell. 2003. *Composition of Secure Multi-Party Protocols, A Comprehensive Study*. Lecture Notes in Computer Science, Vol. 2815. Springer. https://doi.org/10.1007/b13246

[39] Ueli Maurer. 2011. Constructive Cryptography - A New Paradigm for Security Definitions and Proofs. In *Theory of Security and Applications - Joint Workshop, TOSCA 2011, Saarbrücken, Germany, March 31 - April 1, 2011, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 6993)*, Sebastian Mödersheim and Catuscia Palamidessi (Eds.). Springer, 33–56. https://doi.org/10.1007/978-3-642-27375-9_3

[40] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. 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 April 2017, Fort Lauderdale, FL, USA (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Xiaojin (Jerry) Zhu (Eds.). PMLR, 1273–1282. http://proceedings.mlr.press/v54/mcmahan17a.html

[41] Gabriele Oliva, Roberto Setola, Luigi Glielmo, and Christoforos N. Hadjicostis. 2018. Distributed Cycle Detection and Removal. *IEEE Trans. Control. Netw. Syst.* 5, 1 (2018), 194–204. https://doi.org/10.1109/TCNS.2016.2593264

[42] Youyang Qu, Longxiang Gao, Tom H. Luan, Yong Xiang, Shui Yu, Bai Li, and Gavin Zheng. 2020. Decentralized Privacy Using Blockchain-Enabled Federated Learning in Fog Computing. *IEEE Internet Things J.* 7, 6 (2020), 5171–5183. https://doi.org/10.1109/JIOT.2020.2977383

[43] Konrad Rieck, Philipp Trinius, Carsten Willems, and Thorsten Holz. 2011. Automatic analysis of malware behavior using machine learning. *J. Comput. Secur.* 19, 4 (2011), 639–668. https://doi.org/10.3233/JCS-2010-0410

[44] Robert Schmid, Bjarne Pfitzner, Jossekin Beilharz, Bert Arnrich, and Andreas Polze. 2020. Tangle Ledger for Decentralized Learning. In *2020 IEEE International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2020, New Orleans, LA, USA, May 18-22, 2020*. IEEE, 852–859. https://doi.org/10.1109/IPDPSW50202.2020.00144

[45] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. 2018. $D^2$: Decentralized Training over Decentralized Data. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 4855–4863. http://proceedings.mlr.press/v80/tang18a.html

[46] Anh-Tu Tran, The-Dung Luong, Jessada Karnjana, and Van-Nam Huynh. 2021. An efficient approach for privacy preserving decentralized deep learning models based on secure multi-party computation. *Neurocomputing* 422 (2021), 245–262. https://doi.org/10.1016/j.neucom.2020.10.014

[47] Carmela Troncoso, Marios Isaakidis, George Danezis, and Harry Halpin. 2017. Systematizing Decentralization and Privacy: Lessons from 15 Years of Research and Deployments. *Proc. Priv. Enhancing Technol.* 2017, 4 (2017), 404–426. https://doi.org/10.1515/popets-2017-0056

[48] Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. 2017. Decentralized Collaborative Learning of Personalized Models over Networks. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Xiaojin (Jerry) Zhu (Eds.). PMLR, 509–517. http://proceedings.mlr.press/v54/vanhaesebrouck17a.html

[49] Lingyu Wang, Duminda Wijesekera, and Sushil Jajodia. 2002. Cardinality-Based Inference Control in Sum-Only Data Cubes. In *Computer Security - ESORICS 2002, 7th European Symposium on Research in Computer Security, Zurich, Switzerland, October 14-16, 2002, Proceedings (Lecture Notes in Computer Science, Vol. 2502)*, Dieter Gollmann, Günter Karjoth, and Michael Waidner (Eds.). Springer, 55–71. https://doi.org/10.1007/3-540-45853-0_4

[50] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning. In *2019 IEEE Conference on Computer Communications, INFOCOM 2019, Paris, France, April 29 - May 2, 2019*. IEEE, 2512–2520. https://doi.org/10.1109/INFOCOM.2019.8737416

[51] Stanley L. Warner. 1965. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. *J. Amer. Statist. Assoc.* 60, 309 (1965), 63–69. https://doi.org/10.1080/01621459.1965.10480775 arXiv:https://www.tandfonline.com/doi/pdf/10.1080/01621459.1965.10480775

[52] Gary M. Weiss, Jessica L. Timko, Catherine M. Gallagher, Kenichi Yoneda, and Andrew J. Schreiber. 2016. Smartwatch-based activity recognition: A machine learning approach. In *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics, BHI 2016, Las Vegas, NV, USA, February 24-27, 2016*. IEEE, 426–429. https://doi.org/10.1109/BHI.2016.7455925

[53] Lin Xiao and Stephen P. Boyd. 2004. Fast linear iterations for distributed averaging. *Syst. Control. Lett.* 53, 1 (2004), 65–78. https://doi.org/10.1016/J.SYSCONLE.2004.02.022

[54] Bin Yang, Hiroshi Nakagawa, Issei Sato, and Jun Sakuma. 2010. Collusion-resistant privacy-preserving data mining. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, Bharat Rao, Balaji Krishnapuram, Andrew Tomkins, and Qiang Yang (Eds.). ACM, 483–492. https://doi.org/10.1145/1835804.1835867

[55] Valentina Zantedeschi, Aurélien Bellet, and Marc Tommasi. 2020. Fully Decentralized Joint Learning of Personalized Models and Collaboration Graphs. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy] (Proceedings of Machine Learning Research, Vol. 108)*, Silvia Chiappa and Roberto Calandra (Eds.). PMLR, 864–874. http://proceedings.mlr.press/v108/zantedeschi20a.html

[56] Kai Zheng, Wenlong Mou, and Liwei Wang. 2017. Collect at Once, Use Effectively: Making Non-interactive Locally Private Learning Possible. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 4130–4139. http://proceedings.mlr.press/v70/zheng17c.html