University of Warsaw
Faculty of Mathematics, Informatics and Mechanics

Krzysztof Grining

Privacy-Preserving Protocols in Unreliable Distributed Systems

PhD dissertation

Supervisor
dr hab. Marek Klonowski

Co-Supervisor
dr Małgorzata Sulkowska

Faculty of Fundamental Problems of Technology
Wrocław University of Science and Technology

January 2020

Author's declaration:
I hereby declare that this dissertation is my own work.

January 14, 2020 . . . . . .

Krzysztof Grining

Supervisor's declaration:
The dissertation is ready to be reviewed

January 14, 2020 . . . . . .

dr hab. Marek Klonowski

Co-Supervisor's declaration:
The dissertation is ready to be reviewed

January 14, 2020 . . . . . .

dr Małgorzata Sulkowska

Abstract

This thesis concerns chosen problems of privacy preserving data aggregation. It is based on *differential privacy*, which is a mathematically rigorous privacy definition resilient to post-processing. Differential privacy is connected with randomization of the result. The goal is to achieve both sufficient privacy and accuracy. We are interested in practical scenarios, so we consider aggregation in distributed systems with unreliable nodes and untrusted aggregator.

First, we analyse current state-of-the-art solution and show that despite good asymptotical guarantees for the accuracy, in many practical scenarios the errors are unacceptably high. We present our own fault tolerant privacy preserving data aggregation protocol which utilizes limited local communication between nodes. We show that our protocol provides provable level of privacy and far better accuracy even when facing massive failures of nodes.

Next, to make our results useful in wider scenarios, we show how to construct local groups of trust in real-life networks. We consider a distributed system that consists of nodes which need to constitute a huge, connected group in an efficient way, using simple operations and without knowledge of global network topology. We propose and investigate local strategies for constructing large groups of users with low communication and computation overhead. Moreover, we prove some properties of real-life networks while formally assuming that they are generated as a preferential attachment process.

Finally, we took a different approach and focused instead on the privacy definition itself. We look from different perspective at an, already known, relaxation of differential privacy called *noiseless privacy*. It utilizes the randomness in the data, which can either come inherently from the data itself, or model the uncertainty of the Adversary. In contrast to previous work, which focused on asymptotic results, independent data and specific distributions, we give nonasymptotic privacy guarantees for any distribution and a wide class of dependencies. We show a way to combine differential privacy with noiseless privacy and present detailed results which can be easily applied in real-life scenarios of data aggregation.

Keywords: differential privacy, data aggregation, distributed systems, preferential attachment graphs, noiseless privacy

Streszczenie

Przedmiotem tej rozprawy są wybrane problemy agregacji danych z zachowaniem prywatności. Rozprawa jest oparta o *prywatność różnicową* (*differential privacy*), która, w odróżnieniu od wcześniejszych definicji prywatności, jest oparta na formalizmie matematycznym. Prywatność różnicowa wiąże się z odpowiednią randomizacją wyniku. Interesują nas praktyczne scenariusze, więc rozważamy agregacje w rozproszonych systemach z zawodnymi węzłami i niezaufanym agregatorem.

Zaczniemy od przeanalizowania aktualnego rozwiązania problemu i wskazania, że pomimo dobrych asymptotycznych gwarancji dokładności, w wielu praktycznych scenariuszach błędy wynikające z dodanych szumów są nieakceptowalnie duże. Następnie proponujemy skonstruowany przez nas protokół, który wykorzystuje ograniczoną, lokalną komunikację pomiędzy węzłami. Pokazujemy, że nasz protokół zapewnia dowodliwą prywatność oraz jest znacznie dokładniejszy, nawet gdy wiele węzłów jest zawodnych.

Następnie, aby nasze wyniki były użyteczne w szerszej klasie scenariuszy, pokazujemy jak skonstruować lokalne grupy ufających sobie węzłów w realistycznych sieciach. Rozważamy rozproszony system składający się z węzłów, które muszą stworzyć dużą, spójną grupę w sposób efektywny i bez znajomości topologii sieci. Proponujemy i badamy lokalne strategie konstruowania dużych grup z małym narzutem komunikacyjnym i obliczeniowym. Ponadto udowadniamy niektóre własności prawdziwych sieci przy założeniu, że pochodzą z modelu *preferential attachment*.

Na koniec koncentrujemy się na samej definicji prywatności. Rozważamy, znane wcześniej, osłabienie prywatności różnicowej, *noiseless privacy*, wykorzystujące ograniczoną losowość danych. Może ona również modelować niepewność adwersarza. W odróżnieniu od istniejących wyników, które skupiały się na wynikach asymptotycznych, niezależnych danych i konkretnych rozkładach danych, przedstawiamy nieasymptotyczne gwarancje prywatności dla dowolnych rozkładów i szerokiej klasy zależności. Pokazujemy jak połączyć prywatność różnicową z *noiseless privacy* oraz przedstawiamy precyzyjne wyniki, które mogą być łatwo wykorzystane w praktycznych zastosowaniach agregacji danych.

Contents

Chapter 1

Introduction

The problem of preserving privacy when retrieving some function of data has a long history. We are becoming gradually more aware about it with increasing amount of data available. Both the storage capacity and computational power are growing with tremendous speed. This yields more possibilities to analyse data about various populations, which is beneficial, but also about specific individuals, which can be harmful. It may raise some concerns about ones privacy. To make things worse, having more data from various, often seemingly not connected, sources can make privacy breaches even more threatening. The answer to these concerns is *differential privacy*. Unlike previous numerous approaches, which turned out to be flawed and compromised, this is a mathematically rigorous privacy definition resilient to post-processing. This definition of privacy is the central concept of this thesis.

Let us imagine a following problem. There is a set of users and each of them keeps a single value. Analogously, we can think about a database with n records, each corresponding to a specific user. We have to reveal some aggregated statistic (say, the sum of all single values) and preserve the privacy of individuals. In recent years there have been many very promising results, both for the case where the privacy is governed by a trusted, central authority (database curator) and for the case where the data is distributed amongst users who do not trust the data collecting entity.

In this thesis we focus on privacy-preserving protocols in distributed systems. This is due to the importance of such systems and their growing amount, for example mobile users, IoT, smart metering, autonomous vehicles and many others. Most interesting, and practically useful, case is when the system is unreliable, i.e., some of the nodes can fail either randomly or due to some malicious entity. We

propose a various approaches, both by devising a new protocol and by proposing a slightly modified model to improve privacy in such systems.

1.1 Thesis Structure

In Chapter 1 we give a brief introduction to differential privacy, show motivations behind this definition of privacy and present some classic results. Moreover, we recall some of the related literature and mathematical techniques. Next, in Chapter 2 we present fault tolerant privacy preserving data aggregation protocol which utilizes limited local communication between nodes. Furthermore, we analyse current state-of-the-art solution and show that it has unacceptably high errors in practical scenarios. To enhance our results from Chapter 2, in Chapter 3 we propose and investigate local strategies for constructing large groups of users based only on local relations of trust with low communication and computation overhead. Finally, in Chapter 4 we consider a slightly different approach, namely relaxation of differential privacy called noiseless privacy. Chapter 5 is a brief summary of our results.

Most of the content from this thesis is based on published papers. Below we give a short description of these results with specifying the contribution of this thesis' author. The author of this thesis is partially supported by National Science Center (Poland) grant UMO-2018/29/B/ST6/02969.

1. [38] "Practical Fault-Tolerant Data Aggregation", joint work with Marek Klonowski and Piotr Syga. [38] appeared in Proceedings of International Conference on Applied Cryptography and Network Security (ACNS 2016), pp 386-404. This paper is devoted to fault tolerant privacy preserving data aggregation protocols. We analyse the Binary Protocol presented by Chan et al. in [17]. We propose a slightly relaxed model and a precise data aggregation protocol called PAALC, that provides provable level of privacy even when facing massive failures of nodes. The author of this thesis did the analysis of Binary Protocol and co-authored in PAALC protocol design.

2. [39] "On Practical Privacy-Preserving Fault-Tolerant Data Aggregation", extension of [38], joint work with Marek Klonowski and Piotr Syga. [39] appeared in International Journal of Information Security, June 2019, Volume 18, Issue 3, pp 285-304. In this extension we do full analysis of the Binary Protocol presented by Chan et al. in [17]. We analytically show that,

despite being considered state-of-the art for privacy-preserving data aggregation in unreliable distributed systems, it has unacceptable magnitude of errors for most practical applications. We also perform experiments on real data to compare our protocol and the Binary Protocol. The author of this thesis did the full error analysis of the Binary Protocol and performed the experiments.

3. [36] "How to Cooperate Locally to Improve Global Privacy in Social Networks? On Amplification of Privacy Preserving Data Aggregation", joint work with Marek Klonowski and Małgorzata Sulkowska. [36] appeared in the Proceedings of 2017 IEEE Trustcom, pp 464-471. In this paper we propose two protocols which allow to efficiently construct large groups of users based only on local knowledge and trust. We also show that these protocols need very little communication overhead. We present the network as a graph and use so called preferential attachment model, which is known to naturally emerge in real networks. Moreover, we perform experiments on real networks. Both protocols were designed by the author of this thesis.

4. [37] "Stronger Trust and Privacy in Social Networks via Local Cooperation", extension of [36], joint work with Marek Klonowski and Małgorzata Sulkowska. [37] appeared in Journal of Complex Networks. In this extension we propose another local protocol to construct large groups of users and also an approach based on combination of our other two protocols. Moreover, we perform more extensive experiments and formal analysis compared to conference version. Both the protocol design and experiments were done by the author of this thesis.

5. [35] "Towards Extending Noiseless Privacy: Dependent Data and More Practical Approach", joint work with Marek Klonowski. [35] appeared in Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (AsiaCCS 2017), pp 546-560. In this paper we take an already existing relaxation of differential privacy called *noiseless privacy* and further investigate it. We give non-asymptotic results for independent data (previously only asymptotic results were known) and also for dependent data which was not previously analysed in this model. Moreover, we show how to combine standard differential privacy with noiseless privacy to achieve better results. The author of this thesis did the formal analysis of both independent and dependent data cases and proposed the method to combine them with differential privacy.

1.2 Notation and Definitions

$\mathbb{R}$ - set of real numbers

$\mathbb{Z}$ - set of integers

$\mathbb{N}$ - set of natural numbers

sign(x) - the sign function

$\sup_{x \in X}(x)$ - the supremum function

$\max_{x \in X}(x)$ - the maximum function

$\min_{x \in X}(x)$ - the minimum function

$\lfloor x \rfloor$ - the floor function

$\lceil x \rceil$ - the ceiling function

$|x|$ - the absolute value of x

$F_X(t)$ - cumulative distribution function of random variable X

$\binom{n}{k}$ - binomial coefficient

$\mathbb{P}(A)$ - probability of event A

$\mathbb{P}(A|B)$ - probability of event A under condition B

i.i.d. - independent, identically distributed

$X \sim Bin(n, p)$ - random variable X with binomial distribution with n trials and p probability of success, $\mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$

$X \sim N(\mu, \sigma^2)$ - random variable X with normal distribution with mean μ and variance σ^2, with density function $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

$\mathbb{E}(X)$ - expected value of random variable X

$\mathbb{V}ar(X)$ - variance of random variable X

$supp(X)$ - support of random variable X

i - imaginary unit

$G(V, E)$ - graph with set of vertices V and set of edges E

$deg(v)$ - degree of vertex v

$N(v)$ - neighbourhood of vertex v

$A \setminus B$ - set difference

$[n]$ - $\{1, 2, \ldots, n\}$

$G(n, p)$ - Erdös-Renyí graph

whp (with high probability) - we say that an event happens whp if its probability p_n is a function of n and $\lim_{n\to\infty} p_n = 1$

$f(n) \sim g(n)$ - means that $\lim_{n\to\infty} f(n)/g(n) = 1$

$\tilde{\mathrm{O}}\left(f(n)\right)$ - equivalent to $\mathrm{O}\left(f(n)\ln^k\left(f(n)\right)\right)$

1.3 Related Literature

In this section we present some of the most important papers related to this thesis. For more references one can see our papers mentioned in Subsection 1.1.

Until recent years, the definition of anonymity or privacy was somewhat difficult to formulate in a formal way. We want to perform distributed algorithms, use statistical reasoning about population, but with preserving individual privacy. The obvious approach of anonymising the data by removing so called *personally identifying information* like name, surname or ID number from the public data was, and still is, the most popular way of preserving privacy. Unfortunately, it is not sufficient. At first glance it sounds reasonable, because as the data is *sanitized* and *anonymised*, one should not be able to read anything about certain individuals, but only reason about population. However, an obvious question remains - what is, and what is not, a personally identifying information then?

It turned out that such approach to privacy can be easily breached. In [65] the author proposed a simple *linkage attack* where using anonymised medical dataset and public voter dataset he was able to retrieve sensitive data about specific individuals. By comparing ZIP code, date of birth and gender from both datasets he was able to deanonymise sensitive medical data of some individuals. It became

clear that, in reality, there is no such thing as personally identifying information, or rather that everything should be considered as such. The more data we have, the more possibilities of clever linkage attacks exist (see [54]).

This observation led to a slightly more formal concept of anonymity in [55], which describes it as a *state of being not identified within a set of subjects, the "anonymity set"*. Similar approach to privacy in context of databases can be seen in k*-anonymity* metrics ([60, 61, 66]). That is, the privacy is supposed to be preserved as long as each element is revealed in a group of at least k other, identical elements. In this metric as well as some consecutive concepts like ℓ*-diversity* [48] or m*-invariance* [70], the bigger the "anonymity set" is, the stronger the privacy guarantees are. Unfortunately, all these definitions of anonymity, even though they can be considered formal, do not really imply privacy in practice.

In [52] Narayanan and Shmatikov showed that using cleverly devised linkage attacks and appropriate public information one can retrieve sensitive data from large, anonymised and sparse datasets. Their key observation is that the real-life data is often sparse, so any anonymity notion that requires having some number of "similar" objects does not have much practical value. Additionally, in [3, 53, 51] there were also spectacular results concerning deanonymisation of social networks data. The authors were able to deanonymise large social network datasets using just auxilliary data and initial knowledge about a small subset of the network. Privacy attacks appeared also in other areas like collaborative filtering (see [16]) or membership inference in machine learning (see [63]).

An attempt to solve the privacy problem in a more formal way appeared in [21]. Later on, in seminal paper [27] Dwork et al. proposed a formal and rigid notion of privacy, called *differential privacy*. However, its precise formulation in the widely used form appeared in [26]. This was based on the intuition that the only reasonable way of measuring a privacy loss of an individual whose data we possess is by comparing it to a situation where we do not have data of this specific individual. This notion of privacy is the key concept in this thesis.

We also need to mention a crucial extension of *differential privacy* called *local model* (or equivalently *distributed differential privacy*). The foundations of that model were presented in [42]. Some basic algorithms were discussed in [23, 24]. More advanced problem of heavy hitters and histograms revealing in local model were presented in [40] and later refined in [9]. Local model for evolving data was proposed in [41]. Lately a substantially different approach involving shuffling was presented in [18, 31, 4].

Major part of this thesis consists of analysing specific distributed protocols, namely data aggregation, under the regime of differential privacy. Data aggrega-

tion in distributed networks has been thoroughly studied due to its practical importance. Measuring the target environment, aggregating data and raising alarm are arguably the three most important functionalities of distributed sensing networks. With the increased number of personal mobile devices, the aggregation becomes of greatest interest among the three. There are several settings considering data aggregation and they differ in both the abilities and constraints of the nodes performing the aggregation.

The most obvious example of such protocol is revealing e.g. a sum of values of users' data while protecting their privacy. Such aim can be achieved by using combination of cryptography and the standard technique of adding random value (*a noise*) to the aggregated data. Example papers where authors take such approach are for example [62, 57, 29]. It turns out that the bigger the set of individuals contributing to the sum, the less noise has to be added to protect privacy of individuals. Alternatively, having the same level of privacy one can reveal more exact statistics, without risking privacy breaches, if they refer to a bigger set of individuals.

Note that most of protocols described fail to provide the correct output even if only a single user abstains from sending his share of the input. The solution for dynamic networks have been presented in [17]. Approach based on [62] was also focused on more advanced processing of aggregated data (e.g., evaluation and monetization) while protecting privacy of users is discussed in several papers ([1, 30]).

In this thesis we are also analysing various privacy enhancing protocols on real-life networks. Therefore many different papers concerning such networks should be pointed as related work. Since the idea of scale free network modeling appeared, there has been a vast amount of research concerning these, including [2, 6] which laid foundations for scale free modeling of real networks.

Also worth mentioning are papers which provided rigorous mathematical treatment for scale free networks [14, 15, 13]. More recent work on properties of scale free networks include [5, 33]. Throughout our theoretical analysis we use some standard facts concerning random graphs, very comprehensive treatment of these can be found in [12]. We should also mention papers about community structure in large networks [45, 68]. Some empirical results can be found in [19].

Our privacy enhancing protocols are connected with the problem of robustness in complex networks, which has also been widely analyzed. To mention a few papers concerning the robustness and enhancing of robustness in scale free networks we cite [73, 72, 71]. One should also mention [32] wherein authors consider adversarial deletion in scale free graphs and [10], where authors improve

graph robustness by edge modifications.

Another, substantially different, approach to privacy preserving data aggregation is to use some relaxation of differential privacy definition to take into account the adversarial uncertainty. This obviously brings up various problems, like how to measure such uncertainty and how to provide a formal way of quantifying the relaxed version of differential privacy. A concept for such relaxation was introduced in [11] and called *noiseless privacy*. The authors of [11] proposed a new insight considering relaxation of differential privacy which utilizes the uncertainty of the Adversary. This was done in contrast to standard differential privacy, which assumed that the uncertainty has to be injected by a randomized mechanism.

Obviously the notion of differential privacy is quite pessimistic, as we assume that the Adversary knows almost everything. In some cases it makes differential privacy unusable in practice. The necessity to add noise to the final output may render the data completely useless. Imagine situation where we want to do a taxation audit. The aggregator collects the amount of taxes paid by the individuals and then publish their sum. After adding a noise, this sum will be different than the tax due, but now we do not know whether it is because of the noise added, or if there is some tax evasion undergoing.

This might be an extreme example, but nevertheless, a large magnitude of noise (say linear of the size of the data itself) would be prohibitive in most practical situations. One such case is the Binary Protocol from [17], which we thoroughly analyse in Section 2.2, where the magnitude of noises for practical cases is huge, despite good asymptotic properties of the protocol. More comments about the importancy of having appropriate accuracy, especially considering practical deployments of differentially private protocols can be found in [47]

There are also other papers that take similar approach, especially [8, 43]. Both in [8] and [43] the authors proposed a frameworks (called "coupled-worlds privacy" and "Pufferfish", respectively) for specifying privacy definitions utilizing adversarial uncertainty. They could be instantiated in various ways, one of which boils down to noiseless privacy. These are important generalizations of ideas in [11], however the main goal of its authors is extending and generalising privacy definitions. Another paper that is related to differential privacy relaxations is [46], where the authors utilized sampling to enhance privacy. Similarly, in [20] the authors explore inherent privacy properties of cardinality estimators.

1.4 Chosen Mathematical Techniques

In this section we present a few mathematical techniques which are used throughout this thesis.

1.4.1 Characteristic Functions of the Positive Part of a Random Variable

This subsection is devoted to techniques described in [56]. Let X be a real-valued random variable and φ denote its characteristic function, so $\varphi(t) = \mathbb{E}e^{itX}$ for all real t. Denote

$$(J_a\varphi)(t) = \frac{1}{2\pi i}\int\limits_{-\infty}^{\infty} e^{-iua}\varphi(t+u)\frac{du}{u},$$

where a is an arbitrary real number. The integral here is understood in principal-value sense, so that

$$(J_a\varphi)(t) = \lim_{\varepsilon\to 0, A\to\infty}(J_{a,\varepsilon,A}\varphi)(t),$$

where

$$(J_{a,\varepsilon,A}\varphi)(t) := \frac{1}{2\pi i}\left(\int\limits_{\varepsilon}^{A} e^{-iua}\varphi(t+u)\frac{du}{u} + \int\limits_{-A}^{\varepsilon} e^{-iua}\varphi(t+u)\frac{du}{u}\right).$$

Furthermore, we denote

$$J := J_0,$$

and see that

$$J = \frac{i}{2}H,$$

where H is the Hilbert transform given by

$$(H\varphi)(t) := \frac{1}{\pi}\int\limits_{-\infty}^{\infty}\frac{\varphi(s)ds}{t-s}.$$

The author of [56] proves the following

Proposition 1. *(from [56])*

$$(J_a\varphi)(t) = \frac{1}{2}\mathbb{E}\left(e^{itX} sign(X-a)\right).$$

Moreover for all random variables X, a and ε and A such that $0 < \varepsilon < A < \infty$ we have

$$|(J_{a,\varepsilon,A}\varphi)(t)| < 1.$$

First of all, this proposition shows that the integral in $(J_a\varphi)(t)$ always exists as long as φ is characteristic function of some random variable X. Secondly, it yields a useful corollary, which allows to obtain characteristic function of the positive part of a random variable using characteristic function of this variable and its Hilbert transform.

Corollary 1. *(from [56])*

$$\varphi_{X_+}(t) = \mathbb{E}\left(e^{itX_+}\right) = \frac{1}{2}\left[1+\varphi(t)\right] + (J\varphi)(t) - (J\varphi)(0) =$$
$$= \frac{1}{2}[1+\varphi_X(t)] + \frac{1}{2\pi i}\int\limits_{-\infty}^{\infty}\left[\varphi_X(t+u) - \varphi_X(u)\right]\frac{du}{u},$$

where φ is characteristic function of X and $X_+ := max(0, X)$.

1.4.2 Graphs and Preferential Attachment Graphs

In this subsection we will recall both Erdös-Renyí $G(n,p)$ model and preferential attachment graphs model and also some useful facts and theorems.

Let us first recall the well known $G(n,p)$ model (see e.g. [12]).

Definition 1. We say that a graph is from $G(n,p)$ model if it is constructed by connecting n nodes such that each possible edge is included in the graph with probability p independently from every other edge.

In other words, we start with empty graph having n nodes, iterate through all possible pairs and independently place an edge between each pair with probability p. Moreover, we recall an important fact about this model.

Fact 1 (from [12])**.** *Let $G(n,p_n)$ be a random Erdös-Renyí graph on n vertices. If $p_n > \frac{(1+\varepsilon)\log n}{n}$ for some $\varepsilon > 0$ then $G(n,p)$ is whp connected.*

Now we present the second model, namely preferential attachment graph.

Definition 2 (Preferential attachment graph)**.** We say that a graph is from the preferential attachment model with parameter m if it is an effect of the following process. The initial structure is a connected graph on m_0 nodes ($m_0 \geq m \geq 1$). New nodes are added to its structure one at a time. Each new node chooses m existing vertices and attaches to them according to the degree distribution, i.e. the probability that it attaches to a node v is equal to $\frac{deg(v)}{\sum_w deg(w)}$, where the sum runs over all vertices from the present structure.

In this thesis we will also use the following result about preferential attachment graphs which can be found in [33].

Theorem 1. *Let us consider the preferential attachment graph model on n vertices with fixed parameter m. Let $p(l|k)$ denote the probability that a randomly chosen neighbor of a node of degree k will have degree l. Then*

$$p(l|k) \xrightarrow{n\to\infty} \frac{m(k+2)}{kl(l+1)} - \frac{m}{kl}\binom{2m+2}{m+1}\frac{\binom{k+l-2m}{l-m}}{\binom{k+l+2}{l}}.$$

1.4.3 Berry-Esseen Theorem and Stein's Method

In our analysis there is a necessity to find a bound for distance between given mean of a random sample and standard normal distribution. Let us first recall the well known

Fact 2 (Berry-Esseen Theorem)**.** *Let $X_1, \ldots, X_n$ be a sequence of independent random variables. Let $\mathbb{E}X_i = 0$, $\mathbb{E}X_i^2 = \sigma_i^2 > 0$ and $\mathbb{E}|X_i|^3 = \rho_i < \infty$. We denote $\sigma^2 = \sum_{i=1}^n \sigma_i^2$, let F_n denote the cumulative distribution function of $\frac{\sum_{i=1}^n X_i}{\sigma\sqrt{n}}$ and let Φ denote the cumulative distribution function of standard normal distribution. Then*

$$\sup_{x\in\mathbb{R}} |F_n(x) - \Phi(x)| \leqslant \frac{C \cdot \sum_{i=1}^n \rho_i}{(\sum_{i=1}^n \sigma_i^2)^{\frac{3}{2}}},$$

where $C \leqslant 0.5591$ is a constant.

The upper bound for constant C comes from [67].

We also present Stein's Method (see for example [7, 59]), which allows to bound the Kolmogorov distance between two random variables. Firstly, we introduce some notation and facts.

Definition 3 (Dependency Neighborhoods)**.** Let $\{X_1, \ldots, X_n\}$ be a collection of random variables. Their dependency neighborhoods are such $\{N_i\}_{i=1}^n$ that for all N_i we have $N_i \subset [n]$ and X_i is independent of $\{X_k\}_{k \notin N_i}$.

Definition 4 (Kolmogorov Distance)**.** Let X and Y be random variables. We denote their Kolmogorov distance as $d_K(X, Y)$ which is defined as

$$d_K(X, Y) = \sup_{t \in \mathbb{R}} |F_X(t) - F_Y(t)| .$$

Definition 5 (Wasserstein Distance)**.** Let X and Y be random variables. Let μ and ν be their corresponding probability measures. We denote Wasserstein distance as $d_W(X, Y)$ which is defined as

$$d_W(X, Y) = \sup_{h \in \mathcal{H}} \left| \int h(x) d\mu(x) - \int h(x) d\nu(x) \right| ,$$

where $\mathcal{H} = \{h : \mathbb{R} \to \mathbb{R} : |h(x) - h(y)| \leqslant |x - y|\}$.

These are standard probability distribution metrics, their definitions are also given in [59]. We also recall a useful relation between Kolmogorov and Wasserstein distance.

Fact 3 (From [59])**.** *Suppose that a random variable Y has its density bound by some constant C. Then for any random variable X we have*

$$d_K(X, Y) \leqslant \sqrt{2C d_W(X, Y)}.$$

Moreover, if $Y \sim \mathcal{N}(0, 1)$, then for any random variable X we have

$$d_K(X, Y) \leqslant \left(\frac{2}{\pi}\right)^{\frac{1}{4}} \sqrt{d_W(X, Y)}.$$

Lastly, we recall a theorem from [59].

Fact 4 (Theorem 3.6 in [59])**.** *Suppose $X_1, \ldots, X_n$ are random variables such that for every i we have $\mathbb{E}X_i^4 < \infty$, $\mathbb{E}X_i = 0$, $\sigma^2 = \mathbb{V}ar\left(\sum_{i=1}^n X_i\right)$ and define $W = \frac{\sum_{i=1}^n X_i}{\sigma}$. Let the collection $X_1, \ldots, X_n$ have dependency neighborhoods N_i, $i \in [n]$ and also define $D = \max_{1 \leqslant i \leqslant n} |N_i|$. Then, for random variable Z with standard normal distribution we have*

$$d_W(W, Z) \leqslant \frac{D^2}{\sigma^3} \sum_{i=1}^n \mathbb{E}|X_i|^3 + \frac{D^{\frac{3}{2}}\sqrt{28}}{\sigma^2\sqrt{\pi}} \sqrt{\sum_{i=1}^n \mathbb{E}X_i^4}.$$

This fact is obtained by using Stein's method. Note that the Stein's method does not assume anything about joint distribution of dependent subsets, only the size of the greatest dependent subset.

1.5 Differential Privacy Concept

In this section we describe the idea of *differential privacy*, which is a fundamental concept in this thesis, and motivations behind it. Moreover, we recall a few key definitions from [28].

Motivations Let us first describe the motivations behind *differential privacy*. Assume we have a database with data about some population. We want to perform data analysis on it (e.g. calculate average, maximum, median) to infer facts about that population. On the other hand, we want to preserve the privacy of the people from whom we collected the data. The questions we have to ask first are:

- What does *loss of privacy* mean?
- What are our assumptions about the Adversary who wants to breach the privacy of our data?
- What kind of external information does the Adversary have? (data from other sources, public information etc.)
- How to achieve privacy of the data of specific users, yet still give correct (or close to correct) reponses for the queries?

Note that in this section and throughout the whole thesis by *the Adversary* we will mean an abstraction for all the privacy dangers, e.g. external attempt to breach privacy, internal data theft, malicious user and so on. First, naive, idea is to assume that preserving privacy means that it is impossible to infer anything about a specific person based on the responses from our database. This is obviously an approach analogous to *semantic security*. However, this is an infeasible approach. Assume we have a database with height of all male humans and therefore we know the average height of men. Now assume we have access to auxilliary information, from newspaper for example, which says that Mrs X is 10 centimeters taller than an average man in a given population. See that such auxilliary information combined with our knowledge from the database is enough to say how tall exactly Mrs X is. Moreover, her data was not even in the database that we used to breach her

privacy, all we needed was an auxilliary information and data theoretically completely irrelevant to her. Can we really say in such case that database we had lead to a privacy breach? Obviously we want our data to provide meaningful information about some statistics of the population. What about meaningful information about an individual?

Another classic example why we cannot deny such information is as follows. Assume we have performed a study that, for the first time, connected smoking with lung cancer. Mr X has admitted that he is a smoker in last health survey for the insurance company. After publishing the research, his health insurance cost has increased significantly. The insurance company connected the published research with their database and they inferred that Mr X has high risk of lung cancer. Again, it does not matter whether he was, or was not participating in the research data collection, yet meaningful information about him was inferred.

What does differential privacy give then, if we cannot prevent inferring meaningful information about individuals using the data? In [28] there is a perfect answer for that. The authors describe differential privacy as the following promise: "*You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available*". In other words, differential privacy focuses on removing the participation risk - it does not significantly change the outcome of the algorithm whether specific user's data was in the database or not. See that it is a good response for previous toy examples. Mr X participating in the research data for smoking will not significantly change the outcome of the study. That means, when we are approached and asked to participate in a survey which is differentially private, the participation itself will not harm our privacy in any way even though the study itself might infer something about you, that would happen whether you participate or not. Moreover, all these guarantees hold even if the Adversary has any possible auxilliary information about you and **every** other user collides with him.

Definitions In standard differential privacy approach we assume that there exists a trusted *curator* who holds the data of *individuals* in a database D, typically we assume that we have n rows, one for the data of each individual. A *privacy mechanism*, or simply a *mechanism* is an algorithm that takes database D, universe $\mathcal{X}$ of data types (set of all possible database rows) and random bits to produce the output. Let us recall a few definitions from [28]

Definition 6 (Probability Simplex)**.** Given a discrete set B, the *probability simplex*

over B, denoted $\Delta(B)$ is

$$\Delta(B) = \left\{ x \in \mathbb{R}^{|B|} : \forall_i \left(x_i \geqslant 0 \wedge \sum_{i=1}^{|B|} x_i = 1 \right) \right\}.$$

Definition 7 (Randomized Algorithm)**.** A randomized algorithm $\mathcal{M}$ with domain A and discrete range B is associated with a mapping $M : A \to \Delta(B)$. On input $a \in A$, the algorithm $\mathcal{M}$ outputs $\mathcal{M}(a) = b$ with probability $(M(a))_b$ for each $b \in B$.

It is convenient to represent databases as their histograms, namely $x \in \mathbb{N}^{\mathcal{X}}$, where x_i represents number of elements in x of type $i \in \mathcal{X}$. Such representation allows us to recall the following

Definition 8 (Distance between databases)**.** The l_1 distance between two databases x and y is $||x - y||_1$, where

$$||x||_1 = \sum_{i=1}^{|\mathcal{X}|} |x_i|,$$

is the l_1 norm of the database x.

One can easily see that $||x||_1$ measures the size of the database while $||x - y||_1$ measures how many records differ between x and y. Now we are ready to present differential privacy definition, which is a central element of this thesis

Definition 9 (Differential Privacy)**.** A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ)-differentially private, if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leqslant 1$:

$$\mathbb{P}(\mathcal{M}(x) \in \mathcal{S}) \leqslant \exp(\epsilon) \cdot \mathbb{P}(\mathcal{M}(y) \in \mathcal{S}) + \delta,$$

where the probability space is over the coin flips of the mechanism $\mathcal{M}$. If $\delta = 0$ we say that $\mathcal{M}$ is ϵ-differentially private.

One might wonder if randomization of the privacy mechanism is truly necessary. Assume we have a non-trivial (which means there exists at least two inputs which yield different outputs), deterministic privacy mechanism. Then, as such, there exists a query and two databases that yield different output under this query. Changing one row at a time we see that there must exist a pair of databases differing only on single row, for which the query response is different.

One of the most important properties of differential privacy is immunity to post-processing. Let us recall the following

Definition 10 (Immunity to post-processing)**.** Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R$ be a randomized algorithm that is (ϵ, δ)-differentially private. Lef $f : R \to R'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R$ is (ϵ, δ)-differentially private.

Speaking in an informal way, it means that when an analyst gets the output of differentially private algorithm, he cannot in any way increase the privacy loss.

Local Model Now we describe the Local Model, also known as *distributed differential privacy*. Recall that in standard differential privacy model we make a very strong assumption, namely that there is a trusted curator of data, be it either owner of the database or any trusted third party. Generally, we assume that there exists an entity which is allowed to see the data without any obfuscation and that there are secure channels between all users (or other data sources) and this entity. Obviously, this assumption is troublesome for some practical applications. Sometimes we cannot guarantee to the end user that the curator will never be malicious or corrupted. There could be a breach in data security before the appropriate differential privacy technique was applied, which means the Adversary got access to data before any obfuscation. There could also be a breach in the communication channel between user and the database which may allow the Adversary to read his data. All the benefits of differential privacy are gone in such cases, even though the official, published data is sanitized and does not compromise privacy. See that, in some sense, here we consider the curator himself as the Adversary. Despite all the merits of differential privacy mechanisms with trusted curator, in some cases we do not want anyone to get access to our clean data, even the entity that collects and holds it.

The Local Model drops this assumption, which means every user is responsible for adding an appropriate noise to his data (or obfuscating it in other way) before sending it. In other words, we do not have a trusted party which is authorized to collect the real data and then perform some specific actions to preserve privacy (e.g. add noise of appropriate magnitude). The users themselves have to be responsible for securing their privacy by adding noise from some specific distribution, encrypting the noisy value and then sending it to the Aggregator. This problem requires combination of both cryptographic and privacy preserving techniques. In such cases we need both distributed computing and various multiparty computation techniques to ensure the data is collected in a fault-tolerant and cryptographically secure way. Such model makes the privacy problems more complex, yet far more useful in practical scenarios. Of course, there is a price to pay, as most often the bounds for accuracy are worse in the local model than in the general one

and they require more communication due to cryptographic techniques.

To put it more formally we have the following

Definition 11 (Local Randomizer from [28])**.** An ϵ-local randomizer $R : \mathcal{X} \to W$ is an ϵ-differentially private algorithm that takes as input a database of size $n = 1$.

Definition 12 (LR Oracle from [28])**.** An LR oracle $LR_D(\cdot, \cdot)$ takes as input an index $i \in [n]$ and an ϵ-local randomizer R and outputs a random value $w \in W$ chosen according to the distribution $R(x_i)$, where $x_i \in D$ is the element held by the i-th individual.

Definition 13 (Local Algorithm from [28])**.** We will call an algorithm ϵ-local if it accesses the database D via the oracle LR_D, with the following restriction: If $LR_D(i, R_1), \ldots, LR_D(i, R_k)$ are the algorithm's invocations of LR_D on index i, where each R_J is an ϵ_j-local randomizer, then $\epsilon_1 + \ldots + \epsilon_k \leqslant \epsilon$.

See that this formally means that in Local Algorithm, each user has to take care of obfuscating his data and the untrusted aggregator can only collect data via LR Oracle. Note that due to the composability of differential privacy, it is easy to see that ϵ-local algorithms are ϵ-differentially private.

The Local Model is far more practical than standard, so called *centralised*, model. See that nobody other than the data owner has any access to private data and, as long as the protocol is performed in a cryptographically secure way, the data stays private even if the aggregator becomes malicious.

1.6 Basic Differential Privacy Techniques

In this section we give examples of some basic differential privacy techniques such as Randomized Response, Laplace Mechanism and Gaussian Mechanism. These are the common building blocks for more complex algorithms. Note that these definitions and techniques are described for example in [28].

1.6.1 Randomized Response

Let us begin with Randomized Response which is a known folklore mechanism. This is the simplest differential privacy technique which can be easily explained intuitively even to a non-technical person. Assume we conduct a survey, asking people in public a 'yes/no' question, answer to which they probably want to hide, for example about using illegal drugs.

The algorithm for each user goes as follows:

1. Flip a coin
2. If tails, then respond truthfully
3. If heads, then flip a second coin and respond "Yes" if heads and "No" if tails

Claim 1. *(from [28]) Randomized response described above is* $(\ln 3, 0)$*-differentially private*

It is easy to see intuitively, that this algorithm gives so called 'plausible deniability'. Namely that the compromising response could be the effect of random response, not the actual answer. An important feature of this technique is also that it can easily be performed locally. Randomized response could be used for example in conducting privacy-preserving exit polls when we have two candidates for election.

Note that we can easily tune the parameters to either have better ϵ parameter by increasing the chance for a random response, or the opposite, by decreasing the chance for a random response, which yields better accuracy.

1.6.2 Laplace Mechanism

In this subsection we briefly show another basic differential privacy mechanism, known as the Laplace Mechanism which was proposed for the first time in [27]. Here, however, we use more recent definition from [28]. We assume we have a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, so a numeric query (e.g. average or sum of our data). Let us begin with the following

Definition 14 (l_1 - sensitivity)**.** The l_1-sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ is:

$$\Delta f = \max_{x,y \in \mathbb{N}^{\mathcal{X}}; ||x-y||_1 = 1} ||f(x) - f(y)||_1.$$

The l_1 - sensitivity is the magnitude by which a single individual's data can change the numeric query at most. Intuitively, the uncertainty we have to introduce into the response has to be somehow connected with that magnitude. Now let us recall the following definition

Definition 15 (The Laplace Distribution)**.** The Laplace Distribution (centered at 0) with scale b is a distribution with probability density function:

$$Lap(x|b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right).$$

Note that Laplace distribution is a symmetric counterpart of exponential distribution. Having defined both Laplace distribution and l_1-sensitivity we can proceed to define the Laplace Mechanism.

Definition 16 (The Laplace Mechanism, from [28])**.** Given any arbitrary function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, the Laplace mechanism is defined as:

$$\mathcal{M}_L(x, f(\cdot), \epsilon) = f(x) + (Y_1, \ldots, Y_k),$$

where Y_i are i.i.d. random variables drawn from $Lap(\frac{\Delta f}{\epsilon})$.

Most importantly, this mechanism has the following property

Theorem 2. *The Laplace mechanism preserves* $(\epsilon, 0)$*-differential privacy.*

1.6.3 Gaussian Mechanism

We will also present another basic differential privacy mechanism, namely the Gaussian Mechanism. As in previous subsection we assume that we have a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$. Let us begin with the following

Definition 17 (l_2 - sensitivity)**.** The l_2-sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ is:

$$\Delta_2 f = \max_{x,y \in \mathbb{N}^{\mathcal{X}}; ||x-y||_1 = 1} ||f(x) - f(y)||_2.$$

We can define the Gaussian Mechanism

Definition 18 (The Gaussian Mechanism, from [28])**.** Given any arbitrary function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, the Gaussian mechanism is defined as:

$$\mathcal{M}_L(x, f(\cdot), \epsilon) = f(x) + (Y_1, \ldots, Y_k),$$

where Y_i are i.i.d. random variables drawn from $N(0, \sigma^2)$.

Now we present theorem from [25].

Theorem 3 (The Gaussian Mechanism, from [28])**.** *Let* $\varepsilon \in (0, 1)$ *be arbitrary. For* $c^2 > 2ln(1.25/\delta)$*, the Gaussian Mechanism with parameter* $\sigma \geqslant c\Delta_2 f/\varepsilon$ *is* (ε, δ)*-differentially private.*

Chapter 2

Fault Tolerant Privacy-Preserving Data Aggregation Without Trusted Aggregator

Aggregation of data is a fundamental problem that has been approached from different perspectives. In general, this problem has the following setting. There is a set of *users* (we can alternatively call them *nodes*). Each user has some data, for simplification we can think of it as a numeric, or even boolean, value. There is also an entity, called *aggregator*, which calculates some general aggregated statistics (like an average value) based on users' data. From privacy perspective, our goal is to reveal those aggregated statistics while keeping the value of each individual secret, even if the aggregator is untrusted (e.g., tries to learn input of individual users). The general notion is to design a protocol that allows the aggregator to learn a perturbed sum, but no intermediate results (e.g. partial sums).

This chapter is devoted to a fault tolerant privacy preserving data aggregation protocol which utilizes limited local communication between nodes. As a starting point we analyse the Binary Protocol presented by Chan et al. in [17]. Comparing to previous work (see [62]), their scheme guaranteed provable privacy of individuals and fault tolerance. However, we show that despite asymptotic guarantees the error in the Binary Protocol is unacceptably high for practical applications.

Furthermore, we present a precise data aggregation protocol that provides provable level of privacy even when facing massive failures of nodes. Moreover, our protocol requires significantly less computation (limited exploiting of heavy cryptography) than most of fault tolerant aggregation protocols and offers better security guarantees which makes it suitable for systems with limited resources

(e.g. sensor networks). Most importantly, our protocol significantly decreases the error (compared to Binary Protocol). However, to obtain our result we relax the model and allow some limited communication between the nodes. Our approach is a general way to enhance privacy of nodes in networks that allow such limited communication.

2.1 Binary Protocol

In [17] the authors proposed a fault tolerant, privacy preserving data aggregation protocol which has been named *Binary Protocol*. Its purpose is to allow an untrusted Aggregator $\mathbf{AGG}$, to learn the sum of values x_i, $i \in [n]$, where x_i is kept by the i-th user. Their idea is based on earlier work [62], in particular the Block Aggregation protocol. Let us first recall some definitions.

Definition 19. (Symmetric Geometric Distribution). Let $\alpha > 1$. We denote by $Geom(\alpha)$ the symmetric geometric distribution that takes integer values such that the probability mass function at $k \in \mathbb{Z}$ is $\frac{\alpha-1}{\alpha+1} \cdot \alpha^{-|k|}$.

Definition 20. (Diluted Geometric Distribution). Let $\alpha > 1$ and $0 < \beta \leq 1$. A random variable has β-diluted Geometric distribution $Geom^{\beta}(\alpha)$ if with probability β it is sampled from $Geom(\alpha)$, and with probability $1 - \beta$ is set to 0.

The symmetric geometric distribution $Geom(\alpha)$ can be viewed as a discrete version of Laplace distribution. Note however, that we cannot use the Laplace distribution as having discrete values is essential for the cryptographic techniques used in the protocol. The dilution parameter β is the probability that a specific user will add noise from $Geom(\alpha)$. This is done because, intuitively, we want at least one user to add a geometric noise of necessary magnitude, but we do not want too many of these noises to keep the accumulated noise sufficiently small.

Now we briefly recall Block Aggregation protocol from [62].

1. Generate a random secret key sk_i for each of n users as well as an additional sk_0 given to the Aggregator such that $\sum_{i=0}^{n} sk_i = 0$.
2. Each user has data x_i, generates r_i from Diluted Geometric Distribution and encrypts $\tilde{x}_i = x_i + r_i$ using sk_i.
3. Each user sends $\tilde{x}_i$ to the Aggregator.
4. After receiving data from all users Aggregator decrypts the sum using sk_0.

The problem that occurred with Block Aggregation is that whenever a single user fails to deliver their share, it is impossible for the Aggregator to decrypt the desired value.

Binary Protocol presented in [17] addresses the incompleteness of the data by arranging the users in a virtual binary tree. One may visualize each user as a leaf of a binary tree, with all the tree-nodes up to the root being virtual. In order to simulate the tree structure, the users and $\mathbf{AGG}$ are equipped with appropriate secret keys and generate random noises for each of the tree-layer. The first layer consists of the root, second layer consists of its children, and so on. Finally, the $\lceil \log n \rceil + 1^{\text{st}}$ layer consists of the leaves. See that each virtual node corresponds to a subset of users, who are descendants of this specific virtual node. We will call such subsets corresponding to virtual nodes *segments*.

Each user performs Block Aggregation protocol for each of the layers, i.e. $\lceil \log n \rceil + 1$ times. For i-th layer, the noise r_i has diluted geometric distribution with different β_i parameter. Namely, we have $\beta_i = \min\left(\frac{1}{|B_i|} \ln \frac{1}{\delta_0},\, 1\right)$, where $|B_i|$ is the size of segment corresponding to nodes in the layer and $\delta_0 > 0$ is a privacy parameter. If all users present their shares, the problem is reduced to the original Block Aggregation. The Aggregator may decrypt the root-layer block. However, if at least one user fails, all blocks containing this user cannot be decrypted due to lack of necessary secret keys. In order to provide the aggregation of the remaining users the Aggregator has to find a coverage of the tree from the blocks of different layers such that all the remaining users are covered and none of the failed users is included. It is easy to see that such coverage always exists.

Binary Protocol results in $\mathrm{O}\left(n \log n\right)$ communications exchanged in the network and, even more importantly, guarantees $\tilde{\mathrm{O}}\left((\log n)^{\frac{3}{2}}\right)$ error. This notion, however, hides significant constants. In a practical setting, results of [17] are less satisfying than one would expect. The issues concerning the protocol and the resulting error are raised in Section 2.2.

2.2 Analysis of Binary Protocol

In this section we will show that the error magnitude in Binary Protocol is unacceptable for moderate number of participants. Note that in [17] the authors assumed that each user, out of n users, has value $x_i \in \{0, 1\}$, which means that the range of the sum of aggregated data is $[0, n]$. Thus, error of magnitude γn shall be regarded very large even for moderate constant γ.

The authors of [17] have shown and emphasised that the magnitude of error is $o(n)$ asymptotically. However, in practical applications we are also interested in performance of this protocol for moderate values of n, e.g. $n \leqslant 2^{14}$. In this section we will show that for a reasonable range of values of the number of users n and number of failures κ the error is prohibitively large with significant probability. Obviously, as n increases, the Binary Protocol becomes better because of the asymptotic guarantees. However, our aim here is to show, that if the number of participants is moderate or the number of failures is significant (e.g. $\kappa = \log_2(n)$, $\kappa = \lfloor \frac{n}{2^6} \rfloor$) then the accuracy of Binary Protocol is too low to be used in practical applications. Furthermore, if the number of users is quite small (i.e. 2^{10} or less), then even for $\kappa = 5$ the errors generated are unacceptably high. We aim to show a precise magnitude of error in the Binary Protocol. To achieve this, we use subtler method than these presented by the authors of [17].

2.2.1 Analytical Approach

The size of error depends on the number of failed users and the way they are distributed amongst all participants. Let us fix n as the number of participants. Like the authors of [17], throughout this analysis we assume for simplicity that n is a power of 2. However, our reasoning can be generalized to every n. The error generated during the Binary Protocol is the sum of all noises in the aggregated blocks. Throughout this section we will denote $\delta_0 = \frac{\delta}{\lfloor \log_2(n) \rfloor + 1}$, where δ is a privacy parameter. Before we start our analysis, let us recall the following lemma which will be useful later on.

Lemma 1 (Wald's equation (see [50]))**.** *Let $(X_n)_{n\in\mathbb{N}}$ be a sequence of real-valued, i.i.d. random variables and let N be a nonnegative integer-value random variable that is independent of the sequence $(X_n)_{n\in\mathbb{N}}$. Suppose that N and X_n have finite expected values. Then*

$$\mathbb{E}\left(\sum_{i=1}^{N} X_i\right) = E(N)E(X_1).$$

In Binary Protocol each user adds noise from diluted geometric distribution for every layer of the tree. This essentially means that they either add geometric noise with some probability β or no noise at all. Therefore, first we have to show a formula for the expected value of the number of geometric noises added by individual nodes. Note that in the original paper the authors gave only asymptotic

formulas for the number of generated noises. We give an exact formula in the following theorem.

Theorem 4. *Let Y be a random variable which denotes the number of geometric noises added during the Binary Protocol. Let $\kappa > 0$ and fix n as the number of participants. Then, the expected value of random variable Y is given by the following formula:*

$$\mathbb{E}Y = n - \kappa + n \cdot \sum_{i=1}^{\log_2(n)-1} \left(\frac{\binom{n-\frac{n}{2^i}}{\kappa}}{\binom{n}{\kappa}} \cdot (\beta_i - \beta_{i+1}) \right),$$

where $\beta_i = \min\left(\frac{2^i}{n} \ln \frac{1}{\delta_0},\, 1\right)$.

Proof Consider Binary Protocol described in Section 2.1. Assume that $\kappa > 0$ leaves have failed, and they are chosen uniformly at random from all n leaves. Recall from Section 2.1 that $\beta_i = \min\left(\frac{1}{|B_i|} \ln \frac{1}{\delta_0},\, 1\right)$ is the dilution parameter in diluted geometric distribution. We assumed that n is a power of 2, therefore

$$|B_i| = \frac{n}{2^i},$$

because the binary tree is complete. We use random variables X_i to denote the number of segments (on i-th level of the tree) corresponding to a subset of users where noone failed. We call a node an *aggregating node*, if it is a part of the coverage used by the Aggregator to collect data from working users. We will also use random variable X_i^* to denote the number of aggregating nodes on the i-th level of the tree. Let us begin with stating and proving the following

Lemma 2. *Consider Binary Protocol with fixed κ and n. Let X_i^* denote the number of aggregating nodes on the i-th level of the tree. For $i \geqslant 1$ we have*

$$\mathbb{E}X_i^* = \mathbb{E}X_i - 2\mathbb{E}X_{i-1} = 2^i \cdot \left(\frac{\binom{n-\frac{n}{2^i}}{\kappa}}{\binom{n}{\kappa}} - \frac{\binom{n-\frac{n}{2^{i-1}}}{\kappa}}{\binom{n}{\kappa}} \right).$$

Proof First of all, we call a segment in the Binary Protocol tree *clean* if and only if there are no failures in this segment. Each node in the tree corresponds to a specific segment, according to Binary Protocol rules. See that on a certain tree level, all nodes correspond to segments of the same size, noted here by $|B_i|$.

Throughout this reasoning we will call the *root level* 0, children of the root are on level 1 and so on, up to level $\log_2(n)$ which is the *leaves level*.

Value held by each user is, in the end, aggregated in exactly one node, which belongs to some i-th level and corresponds to a specific segment. This user generates a symmetric geometric noise with probability β_i. We want to know the expected value of the number of noises generated throughout the whole protocol. To do this, first we denote the number of *clean* segments of size $|B_i|$ (corresponding to nodes on i-th level of the tree) by a random variable X_i. See that $X_i \in \{0, 1, \ldots, 2^i\}$. Furthermore, we see that:

$$X_i = \sum_{j=1}^{2^i} X_{i,j},$$

where

$$X_{i,j} = \begin{cases} 1, & \text{if segment j on level i has no failures,} \\ 0, & \text{otherwise.} \end{cases}$$

This, and the fact that $\mathbb{E}X_{i,j} = \mathbb{E}X_{i,k}$ for every $j, k \in \{0, \ldots, 2^i\}$, allows us to use linearity of expectation to calculate $\mathbb{E}X_i$:

$$\mathbb{E}X_i = \mathbb{E}\left(\sum_{j=1}^{2^i} X_{i,j}\right) = 2^i \mathbb{E}X_{i,1} = 2^i \cdot \mathbb{P}(X_{i,1} = 1). \tag{2.1}$$

Now see that

$$\mathbb{P}(X_{i,1} = 1) = \frac{\binom{n-|B_i|}{\kappa}}{\binom{n}{\kappa}},$$

and also $|B_i| = \frac{n}{2^i}$, thus plugging these to (2.1) we get

$$\mathbb{E}X_i = 2^i \cdot \frac{\binom{n-\frac{n}{2^i}}{\kappa}}{\binom{n}{\kappa}}. \tag{2.2}$$

See that if node is an *aggregating* one, it means that it corresponds to a *clean* segment, but its parent does not correspond to a *clean* segment. We can see that $X_i^* = X_i - 2X_{i-1}$, where $i \in [\log_2(n)]$. There are X_i *clean* nodes on i-th level but we have to subtract all the *clean* nodes from higher level of the tree multiplicated by 2, because each of these *clean* nodes on a higher level is parent to two nodes on

ith level, which are therefore not *aggregating* nodes, because their parent is *clean*. That gives us

$$\mathbb{E}X_i^* = \mathbb{E}X_i - 2\mathbb{E}X_{i-1} = 2^i \cdot \left(\frac{\binom{n-\frac{n}{2^i}}{\kappa}}{\binom{n}{\kappa}} - \frac{\binom{n-\frac{n}{2^{i-1}}}{\kappa}}{\binom{n}{\kappa}} \right),$$

which completes the proof of this lemma. □

Lemma 2 gives us an explicit formula for $\mathbb{E}X_i^*$. Now we can proceed to calculating the expected value of the number of geometric noises generated during the Binary Protocol.

Let Y_i be a random variable which denotes the number of noises generated on i-th level of the tree. On i-th level we aggregate X_i^* segments, each of these segments have $2^{\log_2(n)-i}$ users and each of these users generates geometric noise with probability β_i. Therefore we have $Y_i = \sum_{j=1}^{2^{\log_2(n)-i} \cdot X_i^*} U_j$, where $U_j \sim Bin(1, \beta_i)$. See that $(U_j)_{n \in \mathbb{N}}$ and $2^{\log_2(n)-i} \cdot X_i^*$ satisfy the assumptions of Lemma 1, so we can apply it and obtain

$$\mathbb{E}Y_i = \mathbb{E}\left(X_i^* \cdot 2^{\log_2(n)-i}\right) \cdot \mathbb{E}U_1 = \mathbb{E}X_i^* \cdot 2^{\log_2(n)-i} \cdot \beta_i.$$

Every user is aggregated only on one level, so if we take a sum over all levels of the tree, we will get all the noises generated during the Binary Protocol. Let Y be a random variable that denotes the number of noises generated. We have

$$Y = \sum_{i=0}^{\log_2(n)} Y_i,$$

and we know that if $\kappa > 0$, then $Y_0 = 0$, because if at least one user has failed, then we cannot aggregate all users in the root of the tree. Furthermore, using linearity of expectation we have

$$\mathbb{E}Y = \sum_{i=1}^{\log_2(n)} \mathbb{E}Y_i = \sum_{i=1}^{\log_2(n)} \mathbb{E}X_i^* \cdot 2^{\log_2(n)-i} \cdot \beta_i.$$

After straightforward calculations we can get

$$\begin{aligned}\mathbb{E}Y &= \sum_{i=1}^{\log_2(n)} \mathbb{E}X_i \cdot 2^{\log_2(n)-i} \cdot \beta_i - \sum_{i=1}^{\log_2(n)} 2\mathbb{E}X_{i-1} \cdot 2^{\log_2(n)-i} \cdot \beta_i = \\ &= \sum_{i=1}^{\log_2(n)} \mathbb{E}X_i \cdot 2^{\log_2(n)-i} \cdot \beta_i - \sum_{i=0}^{\log_2(n)-1} \mathbb{E}X_i \cdot 2^{\log_2(n)-i} \cdot \beta_{i+1} = \\ &= \mathbb{E}X_{\log_2(n)} \cdot \beta_{\log_2(n)} - n\beta_1\mathbb{E}X_0 + \sum_{i=1}^{\log_2(n)-1} \mathbb{E}X_i \cdot 2^{\log_2(n)-i} \cdot (\beta_i - \beta_{i+1}).\end{aligned}$$

Also, as $\kappa > 0$, we have $X_0 = 0$ with probability 1. See also that $\beta_{\log_2(n)} = 1$. These observations yield the following

$$\begin{aligned}\mathbb{E}Y &= \mathbb{E}X_{\log_2(n)} + \sum_{i=1}^{\log_2(n)-1} \mathbb{E}X_i \cdot 2^{\log_2(n)-i} \cdot (\beta_i - \beta_{i+1}) = \\ &= n \cdot \frac{\binom{n-1}{\kappa}}{\binom{n}{\kappa}} + \sum_{i=1}^{\log_2(n)-1} 2^i \cdot \frac{\binom{n-\frac{n}{2^i}}{\kappa}}{\binom{n}{\kappa}} \cdot 2^{\log_2(n)-i} \cdot (\beta_i - \beta_{i+1}) = \\ &= n - \kappa + n \cdot \sum_{i=1}^{\log_2(n)-1} \left(\frac{\binom{n-\frac{n}{2^i}}{\kappa}}{\binom{n}{\kappa}} \cdot (\beta_i - \beta_{i+1}) \right).\end{aligned}$$

This gives us a formula for $\mathbb{E}Y$ and completes the proof of this theorem. □

Now we show a lower bound for the expected number of noises for limited range of n. We present it in the following

Corollary 2. *Let $2^4 \leqslant n \leqslant 2^{21}$ and $\delta = 0.05$, then $\mathbb{E}Y$ has a following lower bound:*

$$\mathbb{E}Y \geqslant n - \kappa - n \cdot \left(e^{-\frac{8\kappa}{n}} + \frac{\ln(\frac{\log_2(n)+1}{\delta})}{8} \cdot \left(e^{-\frac{16\kappa}{n}} - e^{-\frac{8\kappa}{n}} \right) \right).$$

Proof We fix $\delta = 0.05$. First observe that for $2^4 \leqslant n \leqslant 2^{21}$ we have $\beta_{\log_2(n)} = \beta_{\log_2(n)-1} = \beta_{\log_2(n)-2} = 1$, as for these levels we have $\frac{1}{|B_i|} \cdot \ln(\log_2(n) + 1) > 1$. Therefore users aggregated in segments of length 1 and 2 generate noise with probability 1. Furthermore, for $i \leqslant (\log_2(n) - 3)$ we have $\beta_i < 1$. Also, for

$i \leqslant (\log_2(n) - 4)$ we have $\frac{\beta_{i+1}}{\beta_i} = \frac{|B_i|}{|B_{i+1}|} = 2$. Another observation is that we can get an upper bound for $\frac{\binom{n-\frac{n}{2^i}}{\kappa}}{\binom{n}{\kappa}}$ in a following way

$$\frac{\binom{n-\frac{n}{2^i}}{\kappa}}{\binom{n}{\kappa}} = \frac{(n \cdot \frac{2^i-1}{2^i}) \cdot (n \cdot \frac{2^i-1}{2^i} - 1) \cdot \ldots \cdot (n \cdot \frac{2^i-1}{2^i} - \kappa + 1)}{n \cdot (n-1) \cdot \ldots \cdot (n - \kappa + 1)} =$$

$$= \left(\frac{2^i - 1}{2^i}\right)^\kappa \cdot \frac{n \cdot (n - \frac{2^i}{2^i-1}) \cdot \ldots \cdot (n - (\kappa - 1) \cdot \frac{2^i}{2^i-1})}{n \cdot (n-1) \cdot \ldots \cdot (n - \kappa + 1)} \leqslant$$

$$\leqslant \left(\frac{2^i - 1}{2^i}\right)^\kappa = \left(1 - \frac{1}{2^i}\right)^\kappa \leqslant e^{-\frac{\kappa}{2^i}},$$

where the last inequality comes from the fact that $(1 - x) \leqslant e^{-x}$. We can use all these observations to obtain a lower bound. Let $\beta^* = \ln\left(\frac{\log_2(n)+1}{\delta}\right)$. Then

$$\mathbb{E}Y = n - \kappa + n \cdot \sum_{i=1}^{\log_2(n)-1} \left(\frac{\binom{n-\frac{n}{2^i}}{\kappa}}{\binom{n}{\kappa}} \cdot (\beta_i - \beta_{i+1})\right) =$$

$$= n - \kappa - n \cdot \left(\sum_{i=1}^{\log_2(n)-4} \left(\frac{\binom{n-\frac{n}{2^i}}{\kappa}}{\binom{n}{\kappa}} \cdot \beta_i\right) + \frac{\binom{n-8}{\kappa}}{\binom{n}{\kappa}} \cdot \left(1 - \beta_{\log_2(n)-3}\right)\right) \geqslant$$

$$\geqslant n - \kappa - n \cdot \left(\sum_{i=1}^{\log_2(n)-4} \left(e^{-\frac{\kappa}{2^i}} \cdot \beta_i\right) + e^{\frac{8\kappa}{n}} \cdot \left(1 - \beta_{\log_2(n)-3}\right)\right) \geqslant$$

$$\geqslant n - \kappa - n \cdot \left(\sum_{i=1}^{\log_2(n)-4} \left(e^{-\frac{\kappa}{2^{\log_2(n)-4}}} \cdot \beta_i\right) + e^{\frac{8\kappa}{n}} \cdot \left(1 - \beta_{\log_2(n)-3}\right)\right) =$$

$$= n - \kappa - n \cdot \left(e^{-\frac{16\kappa}{n}} \cdot \frac{\beta^*}{n} \cdot \sum_{i=1}^{\log_2(n)-4} \left(2^i\right) + e^{\frac{8\kappa}{n}} \cdot \left(1 - \beta_{\log_2(n)-3}\right)\right) =$$

$$= n - \kappa - n \cdot \left(e^{-\frac{16\kappa}{n}} \cdot \frac{\beta^*}{n} \cdot \left(\frac{n}{8} - 2\right) + e^{\frac{8\kappa}{n}} \cdot \left(1 - \frac{\beta^*}{8}\right)\right) \geqslant$$

$$\geqslant n - \kappa - n \cdot \left(e^{-\frac{16\kappa}{n}} \cdot \frac{\beta^*}{8} + e^{\frac{8\kappa}{n}} \cdot \left(1 - \frac{\beta^*}{8}\right)\right) =$$

$$= n - \kappa - n \cdot \left(e^{-\frac{8\kappa}{n}} + \frac{\beta^*}{8} \cdot \left(e^{-\frac{16\kappa}{n}} - e^{-\frac{8\kappa}{n}}\right)\right).$$

Which gives lower bound for $\mathbb{E}Y$ and finishes the proof of this corollary. $\square$

Note that if $n < 2^4$ then we have $\beta_i = 0$, which means that every remaining user has to add noise (even if there are no failures, i.e $\kappa = 0$). There is no need to give a lower bound in that case, because then the number of noisy inputs is exactly $n - \kappa$. Note also that even though we fixed a specific δ that is used broadly in various papers (including [17]), similar reasoning can be made for different values of δ.

We use this result to show the following

Example 1. Fix $\delta = 0.05$. We will plot the lower bound for the fraction of nodes that added noise in Binary Protocol, i.e. lower bound for $\frac{\mathbb{E}Y}{n}$, using Corollary 2. In Figure 2.1 we assumed $\kappa = \log_2(n)$ failures. See that for moderate number of nodes, the fraction of nodes that actually added noise from geometric distribution is linear in n. In Figure 2.2 we set $\kappa = \frac{n}{2^6}$ which is still less than 2% failures. The fraction of users that generated noise is over 17%. Recall that ideally there should be only a single noise or a constant number of those.

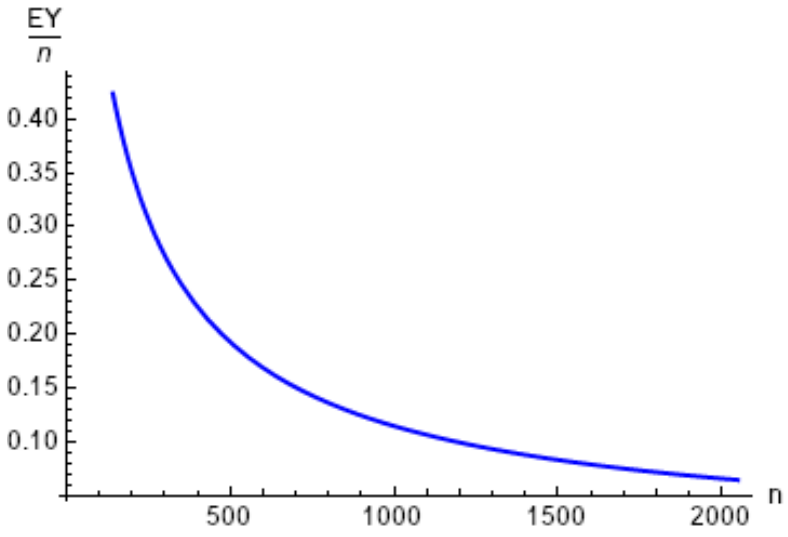


Figure 2.1: Lower bound for $\frac{\mathbb{E}Y}{n}$ in Binary Protocol with $\delta = 0.05$, $\kappa = \log_2(n)$.

It can easily be seen in the Example 1 that even if the number of failures is very small (i.e. less than 2% users with failures), the number of noises generated is linear in n for realistic number of nodes. Note that it does not yet mean that the size of the error is linear, because the noises could cancel each other out to some extent.

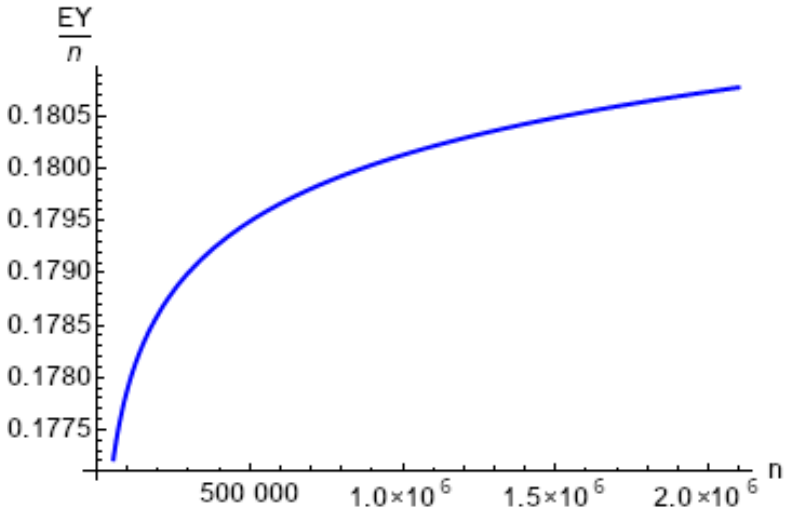


Figure 2.2: Lower bound for $\frac{\mathbb{E}Y}{n}$ in Binary Protocol with $\delta = 0.05$, $\kappa = \frac{n}{2^6}$.

Having an exact formula and also a lower bound for the expected number of noises generated, we can calculate the error. Let us assume that we have m noises generated. Recall that each of them comes from symmetric geometric distribution $Geom(\alpha)$ with $\alpha > 1$. We denote the sum of all noises as Z. One can easily see that $\mathbb{E}Z = 0$ due to symmetry of distribution. However the expected additional error i.e., $\mathbb{E}|Z|$ might be, and we will show that it often is, quite large.

Theorem 5. *Let m denote the number of noises generated in Binary Protocol, each coming from $Geom(\alpha)$ distribution for fixed α. Then let $Z = \sum_{i=1}^{m} Z_i$ be a random variable which denotes the sum of generated noises. We have*

$$\mathbb{E}|Z| = \int_0^\infty \frac{4 \cdot \alpha \cdot m \cdot \sin t \cdot (\alpha - 1)^{2m}}{t \cdot \pi \cdot (\alpha^2 - 2\alpha \cos t + 1)^{m+1}} dt.$$

Proof Let $\varphi_{Z_i}(t)$ denote the characteristic function of Z_i. We have

$$\varphi_{Z_i}(t) = \frac{(\alpha - 1)^2}{\alpha^2 - \alpha(e^t + e^{-t}) + 1} = \frac{(\alpha - 1)^2}{\alpha^2 - 2\alpha \cos t + 1}.$$

Let $\varphi_Z(t)$ denote the characteristic function of Z. As Z_i are i.i.d. random variables, we get

$$\varphi_Z(t) = (\varphi_{Z_1})^m = \left(\frac{(\alpha - 1)^2}{\alpha^2 - 2\alpha \cos t + 1} \right)^m.$$

We use techniques described in Subsection 1.4.1 to calculate the expected value of $|Z|$. We denote $Z_+ = \max(0, Z)$ and $Z_- = \max(0, -Z)$. Using Corollary 1 we obtain

$$\varphi_{Z_+}(t) = \mathbb{E}e^{itZ_+} = \frac{1}{2}[1 + \varphi_Z(t)] + \frac{1}{2\pi i} \int\limits_{-\infty}^{\infty} [\varphi_Z(t+u) - \varphi_Z(u)] \frac{du}{u}.$$

The integral is understood in the principal value sense (see Subsection 1.4.1). Recall that Z is symmetric, so

$$|Z| = Z_+ + Z_- = Z_+ + (-Z_+) = 2Z_+.$$

Furthermore, we have

$$\mathbb{E}|Z| = 2\mathbb{E}Z_+ = 2\frac{\varphi'_{Z_+}(0)}{i}. \tag{2.3}$$

We have to calculate the derivative of $\varphi_{Z_+}(t)$ at 0. It can be done in the following way

$$\begin{aligned} \varphi'_{Z_+}(0) &= \frac{\varphi'_Z(0)}{2} + \frac{d}{dt}\left(\frac{1}{2\pi i} \int\limits_{-\infty}^{\infty} [\varphi_Z(t+u) - \varphi_Z(u)] \frac{du}{u}\right)(0) \\ &= \frac{1}{2\pi i}\left(\int\limits_{-\infty}^{\infty} [\varphi'_Z(t+u)] \frac{du}{u}\right)(0) = \frac{1}{2\pi i} \int\limits_{-\infty}^{\infty} [\varphi'_Z(u)] \frac{du}{u}. \end{aligned} \tag{2.4}$$

Since Z is symmetric then $\varphi'_Z(0) = 0$. Moreover, because $\mathbb{E}Z$ exists and is finite, then $\mathbb{E}|Z|$ also has to exist. Therefore the integral has to be finite, so we can use Lebesgue theorem to swap the order of derivation and integration. We can derive $\varphi_Z(t)$ which yields the following

$$\varphi'_Z(t) = \frac{-2 \cdot \alpha \cdot m \cdot \sin t \cdot (\alpha - 1)^{2m}}{(\alpha^2 - 2\alpha \cos t + 1)^{m+1}}. \tag{2.5}$$

Combining (2.3), (2.4), (2.5) and observing that $\frac{\varphi'_Z(t)}{t}$ is an even function, we obtain the following formula for $\mathbb{E}|Z|$

$$\mathbb{E}|Z| = \int\limits_0^{\infty} \frac{4 \cdot \alpha \cdot m \cdot \sin t \cdot (\alpha - 1)^{2m}}{t \cdot \pi \cdot (\alpha^2 - 2\alpha \cos t + 1)^{m+1}} dt,$$

which completes the proof of this theorem. □

We also show a lower bound for $\mathbb{E}|Z|$ in a following

Fact 5. *For fixed n and ϵ, we denote $\alpha = \frac{\epsilon}{\log_2(n)+1}$ and $m = \gamma n$, for $\gamma \in (0, 1]$. Then, provided that $\sqrt{\frac{\pi(\alpha-1)^2}{4\alpha m}} \geqslant 2\pi - 5$ we have*

$$\mathbb{E}|Z| \geqslant c_{n,\epsilon} \cdot \sqrt{\gamma} \cdot \frac{\log_2(n) \cdot \sqrt{n}}{\epsilon\sqrt{\pi}} - 0.1 \ ,$$

where $c_{n,\epsilon}$ is a constant, which is at least 1.4 for moderate values of n and ϵ.

Proof Let us define $\omega(t)$

$$\omega(t) = \frac{4 \cdot \alpha \cdot m \cdot \sin t \cdot (\alpha - 1)^{2m}}{\pi \cdot (\alpha^2 - 2\alpha \cos t + 1)^{m+1}}.$$

We have

$$\mathbb{E}|Z| = \int_0^\infty \frac{4 \cdot \alpha \cdot m \cdot \sin t \cdot (\alpha - 1)^{2m}}{t \cdot \pi \cdot (\alpha^2 - 2\alpha \cos t + 1)^{m+1}} dt = \int_0^\infty \frac{\omega(t)}{t}.$$

One can easily see that $\omega(t)$ is periodic with period 2π. We can therefore consider splitting the integral into $[2k\pi, 2(k+1)\pi]$ intervals and try to find an accurate lower bound for this integral. We have

$$\mathbb{E}|Z| = \sum_{k=0}^{\infty} \left(\int_{2k\pi}^{2(k+1)\pi} \frac{\omega(t)}{t} dt \right).$$

Consider any of these integrals for $k \geqslant 0$

$$\int_{2k\pi}^{2(k+1)\pi} \frac{\omega(t)}{t} dt \geqslant 0. \tag{2.6}$$

We will now explain why this inequality holds. First, observe that function $\omega(t)$ is an odd function on the interval $[2k\pi, 2(k+1)\pi]$. One can easily see, that $\omega(t)$ is positive on $[2k\pi, 2k\pi + \pi]$ and negative on $[2k\pi + \pi, 2(k+1)\pi]$. Furthermore, the absolute value of $\frac{\omega(t)}{t}$ is greater on the first half of the interval, because of the

decreasing factor $\frac{1}{t}$. This yields (2.6), which is true for all these intervals, and we will use it for all $k > 0$, so that leaves us with

$$\mathbb{E}|Z| = \sum_{k=0}^{\infty} \left(\int_{2k\pi}^{2(k+1)\pi} \frac{\omega(t)}{t} dt \right) \geqslant \int_{0}^{2\pi} \frac{\omega(t)}{t} dt.$$

We could use the lower bound (2.6), however there is no point using it on the whole interval, because we would obtain trivial inequality $\mathbb{E}|Z| \geqslant 0$. It requires slightly subtler handling. Clearly, we could use (2.6) for any interval of type $[\pi - x, \pi + x]$, for $x \in [0, \pi]$. This yields the following

$$\begin{aligned}\mathbb{E}|Z| &\geqslant \int_{0}^{2\pi} \frac{\omega(t)}{t} dt = \int_{0}^{\eta_{\alpha,m}} \frac{\omega(t)}{t} dt + \int_{\eta_{\alpha,m}}^{2\pi-\eta_{\alpha,m}} \frac{\omega(t)}{t} dt + \int_{2\pi-\eta_{\alpha,m}}^{2\pi} \frac{\omega(t)}{t} dt \geqslant \\ &\geqslant \int_{0}^{\eta_{\alpha,m}} \frac{\omega(t)}{t} dt + \int_{2\pi-\eta_{\alpha,m}}^{2\pi} \frac{\omega(t)}{t} dt,\end{aligned}$$

which is true for every $\eta_{\alpha,m} \in [0, \pi]$. Now see that if $\eta_{\alpha,m} < \frac{\pi}{2}$, we can bound the first integral in a following way

$$\int_{0}^{\eta_{\alpha,m}} \frac{4\alpha m \cdot \sin t \cdot (\alpha - 1)^{2m}}{t \cdot \pi \cdot (\alpha^2 - 2\alpha \cos t + 1)^{m+1}} dt \geqslant \int_{0}^{\eta_{\alpha,m}} \frac{4\alpha m \cdot \cos t \cdot (\alpha - 1)^{2m}}{\pi \cdot (\alpha^2 - 2\alpha \cos t + 1)^{m+1}} dt, \tag{2.7}$$

which follows from the fact that $\cos t \leqslant \frac{\sin t}{t}$ for $t \in [0, \frac{\pi}{2})$. Furthermore

$$\int_{2\pi-\eta_{\alpha,m}}^{2\pi} \frac{4\alpha m \cdot \sin t \cdot (\alpha - 1)^{2m}}{t \cdot \pi \cdot (\alpha^2 - 2\alpha \cos t + 1)^{m+1}} dt \geqslant \int_{2\pi-\eta_{\alpha,m}}^{2\pi} \frac{4\alpha m \cdot \sin t \cdot (\alpha - 1)^{2m}}{t \cdot \pi \cdot (\alpha - 1)^{2m+2}} dt, \tag{2.8}$$

which comes from plugging 1 instead of $\cos t$, which makes the function greater in terms of absolute value, but as it is negative on this interval, it yields a lower bound. In (2.8) we have, in fact, an integral of $\frac{\sin t}{t}$ multiplied by a constant depending on α and m. There also still remains a problem of choosing $\eta_{\alpha,m}$. First we can observe that, for small enough $\eta_{\alpha,m}$ we have

$$\int_{2\pi-\eta_{\alpha,m}}^{2\pi} \frac{\sin t}{t} dt \geqslant -\frac{\eta_{\alpha,m}^2}{10}.$$

Obviously this holds for $\eta_{\alpha,m} = 0$. Let $Si(x)$ denote the antiderivative of $\frac{\sin x}{x}$. After derivating left side we obtain

$$\frac{d\left(Si(2\pi) - Si(2\pi - \eta_{\alpha,m})\right)}{d\eta_{\alpha,m}} = -\frac{d\left(Si(2\pi - \eta_{\alpha,m})\right)}{d\eta_{\alpha,m}} = \frac{\sin\left(2\pi - \eta_{\alpha,m}\right)}{2\pi - \eta_{\alpha,m}} =$$
$$= -\frac{\sin(\eta_{\alpha,m})}{2\pi - \eta_{\alpha,m}} \geqslant -\frac{\eta_{\alpha,m}}{2\pi - \eta_{\alpha,m}}.$$

Derivating the right side yields $-0.2\eta_{\alpha,m}$. We can check when the left side is greater than the right side

$$-\frac{\eta_{\alpha,m}}{2\pi - \eta_{\alpha,m}} \geqslant -0.2\eta_{\alpha,m},$$

which is true when

$$\eta_{\alpha,m} \leqslant 2\pi - 5.$$

So for $\eta_{\alpha,m} \leqslant (2\pi - 5)$ we have

$$\int_{2\pi - \eta_{\alpha,m}}^{2\pi} \frac{\sin t}{t} dt \geqslant -\frac{\eta_{\alpha,m}^2}{10}.$$

Now we pick $\eta_{\alpha,m}$ so that

$$-0.1\eta_{\alpha,m}^2 \cdot \frac{4\alpha m}{\pi(\alpha - 1)^2} = -0.1.$$

That gives us

$$\eta_{\alpha,m} = \sqrt{\frac{\pi(\alpha - 1)^2}{4\alpha m}}.$$

Plugging it all to our formula for expected magnitude of noise yields

$$\mathbb{E}|Z| \geqslant \int_0^{\eta_{\alpha,m}} \frac{4 \cdot a \cdot m \cdot \cos t \cdot (\alpha - 1)^{2m}}{\pi \cdot (\alpha^2 - 2\alpha \cos t + 1)^{m+1}} dt - 0.1.$$

We are now interested in the lower bound for this integral. One can see that

$$\int_0^{\eta_{\alpha,m}} \frac{4 \cdot a \cdot m \cdot \cos t \cdot (\alpha - 1)^{2m}}{\pi \cdot (\alpha^2 - 2\alpha \cos t + 1)^{m+1}} dt \geqslant \int_0^{\eta_{\alpha,m}} \frac{4 \cdot a \cdot m \cdot \cos(\eta_{\alpha,m}) \cdot (\alpha - 1)^{2m}}{\pi \cdot (\alpha^2 - 2\alpha \cos t + 1)^{m+1}} dt.$$

This inequality is just plugging the smallest possible value of cosine on this interval. Furthermore, we have

$$\int_0^{\eta_{\alpha,m}} \frac{4\alpha m \cdot \cos(\eta_{\alpha,m}) \cdot (\alpha-1)^{2m}}{\pi\left(\alpha^2 - 2\alpha\cos t + 1\right)^{m+1}} dt \geqslant \int_0^{\eta_{\alpha,m}} \frac{4\alpha m \cdot \left(1-\frac{\eta_{\alpha,m}^2}{2}\right) \cdot (\alpha-1)^{2m}}{\pi\left(\alpha^2 - 2\alpha\cdot\left(1-\frac{t^2}{2}\right) + 1\right)^{m+1}} dt.$$

This bound comes from the fact that $\cos t \geqslant \left(1-\frac{t^2}{2}\right)$. Let us call the integrand function $g(t)$. This function has a following anti-derivative $G(t)$:

$$G(t) = \frac{4(\alpha-1)^{2m-2}\alpha m t \left(1+\frac{\alpha t^2}{(\alpha-1)^2}\right)^m \left(1-\frac{\eta_{\alpha,m}^2}{2}\right) {}_2F_1\left(\frac{1}{2}, 1+m, \frac{3}{2}, -\frac{\alpha\cdot t^2}{(\alpha-1)^2}\right)}{\left(\alpha^2+\alpha(t^2-2)+1\right)^m \cdot \pi},$$

where the ${}_2F_1(a,b,c,z)$ denotes ordinary hypergeometric function (see [69]). One can easily see, that $G(0)=0$. That leaves us with

$$\mathbb{E}|Z| \geqslant G(\eta_{\alpha,m}) - 0.1\ .$$

$G(\eta_{\alpha,m})$ is quite complicated, but we can greatly simplify it. Let us begin with taking some of the $G(\eta_{\alpha,m})$ factors

$$\begin{aligned}\frac{(\alpha-1)^{2m-2}\cdot\left(1+\frac{\alpha\cdot\eta_{\alpha,m}^2}{(\alpha-1)^2}\right)^m}{\left(\alpha^2+\alpha\cdot(\eta_{\alpha,m}^2-2)+1\right)^m} &= \frac{(\alpha-1)^{-2}\cdot\left(1+\frac{\alpha\cdot\eta_{\alpha,m}^2}{(\alpha-1)^2}\right)^m}{\left(\frac{\alpha^2}{(\alpha-1)^2}+\frac{\alpha}{(\alpha-1)^2}\cdot(\eta_{\alpha,m}^2-2)+\frac{1}{(\alpha-1)^2}\right)^m} = \\ &= \frac{(\alpha-1)^{-2}\cdot\left(1+\frac{\alpha\cdot\eta_{\alpha,m}^2}{(\alpha-1)^2}\right)^m}{\left(1+\frac{\alpha\cdot\eta_{\alpha,m}^2}{(\alpha-1)^2}\right)^m} = (\alpha-1)^{-2}.\end{aligned}$$

Furthermore, we can expand ${}_2F_1(a,b,c,z)$ into Taylor series around 0 in a following way:

$${}_2F_1\left(\frac{1}{2}, 1+m, \frac{3}{2}, -\frac{\alpha t^2}{(\alpha-1)^2}\right) = 1 - \frac{\alpha(m+1)t^2}{3(\alpha-1)^2} + O(t^4) \geqslant 1 - \frac{\alpha(m+1)\eta_{\alpha,m}^2}{3(\alpha-1)^2}.$$

Using these two observations we obtain

$$G(\eta_{\alpha,m}) \geqslant \frac{4(\alpha-1)^{-2}\cdot\alpha\cdot m\cdot\eta_{\alpha,m}\cdot\left(1-\frac{\eta_{\alpha,m}^2}{2}\right)\cdot\left(1-\frac{\alpha\cdot(m+1)\cdot\eta_{\alpha,m}^2}{3\cdot(\alpha-1)^2}\right)}{\pi}.$$

We can further simplify this by recalling that $\alpha = e^{\frac{\epsilon}{\log_2(n)+1}}$ and $m = \gamma n$ and observing that the expression $\left(1 - \frac{\eta^2_{\alpha,m}}{2}\right) \cdot \left(1 - \frac{\alpha \cdot (m+1) \cdot \eta^2_{\alpha,m}}{3 \cdot (\alpha-1)^2}\right)$ is increasing with n. Let us call this value c^*_n. We can fix this for the smallest n that we want to consider. See that, for example, for $n \geqslant 2^7$ we have $c^*_n \geqslant 1.43$. This leaves us with

$$\begin{aligned} G(\eta_{\alpha,m}) &\geqslant \frac{4c^*_n \cdot (\alpha-1)^{-2} \cdot \alpha \cdot m \cdot \eta_{\alpha,m}}{\pi} = \\ &= \frac{4c^*_n \cdot (\alpha-1)^{-2} \cdot \alpha \cdot m \cdot \sqrt{\frac{\pi(\alpha-1)^2}{4\alpha m}}}{\pi} = \\ &= \frac{2c^*_n \cdot \sqrt{\alpha \cdot m}}{\sqrt{\pi} \cdot (\alpha-1)} \geqslant \frac{2c^*_n \cdot \sqrt{m}}{\sqrt{\pi} \cdot (\alpha-1)} = \frac{2c^*_n \cdot \sqrt{\gamma n}}{\sqrt{\pi} \cdot (e^{\frac{\epsilon}{\log_2(n)+1}} - 1)} \geqslant \\ &\geqslant \frac{2c^*_n \cdot \sqrt{\gamma n}}{\sqrt{\pi} \cdot (e^{\frac{\epsilon}{\log_2(n)}} - 1)} \geqslant \frac{\xi \log_2(n) \cdot 2c^*_n \cdot \sqrt{\gamma n}}{\epsilon\sqrt{\pi}}, \end{aligned}$$

where ξ is such that $e^{\xi \cdot x} \leqslant (1+x)$ for $x = (\frac{1}{2\log_2(n)})$. For example, in case we have $\epsilon = 0.5$ and $n \geqslant 2^7$ it suffices to take $\xi = 0.96$. In the end we have

$$G(\eta_{\alpha,m}) \geqslant c_{n,\epsilon} \cdot \sqrt{\gamma} \cdot \frac{\log_2(n) \cdot \sqrt{n}}{\epsilon\sqrt{\pi}},$$

where $c_{n,\epsilon} = 2\xi c^*_n$ which is, for moderate n and ϵ, greater than 1.4. In fact, for $\epsilon = 0.5$ and $n \geqslant 2^7$ it is greater than 2. In the end we have

$$\mathbb{E}|Z| \geqslant c_{n,\epsilon} \cdot \sqrt{\gamma} \cdot \frac{\log_2(n) \cdot \sqrt{n}}{\epsilon\sqrt{\pi}} - 0.1\ ,$$

which completes the proof of this fact. □

Using Fact 5 we can obtain a following

Example 2. Consider Binary Protocol for $\delta = 0.05$, $\epsilon = 0.5$, $n \leqslant 2^{10}$ and $\kappa = \log_2(n)$. Let $|Z|$ be the absolute value of all noises aggregated during this protocol. We have $\mathbb{E}|Z| \geqslant 0.15 \cdot n$. Moreover, if we take $\kappa = \frac{n}{2^6}$ and $2^6 \leqslant n \leqslant 2^{12}$ we have $\mathbb{E}|Z| \geqslant 0.12 \cdot n$.

This is an immediate result from the Fact 5, we can see that $\frac{\mathbb{E}|Z|}{n}$ is a decreasing function of n. Therefore it is enough to plug $n = 2^{10}$ into lower bound for $\mathbb{E}|Z|$ for the first part of the corollary and $n = 2^{12}$ for the second part of the corollary.

This clearly shows that even if we consider the lower bound for the number of noises and their magnitude, the Binary Protocol is far from perfect for many realistic scenarios, i.e. when the number of participants is moderate. Even worse conclusions are drawn in Subsection 2.2.2, where we use the exact formulas given in theorems 4 and 5 to numerically analyze the errors generated in this protocol.

2.2.2 Numerical Approach

In Subsection 2.2.1 we gave both exact formulas and lower bounds for the number of noises generated and their sum. Here we show that the errors generated are, in fact, even larger. We use the exact formulas to precisely calculate the errors numerically. First let us consider the case where $n \leqslant 2^{10}$, $\kappa = \lfloor \log_2(n) \rfloor$, and privacy parameters are $\epsilon = 0.5$, $\delta = 0.05$. See Figure 2.3. It clearly shows that the error magnitude in Binary Protocol is, in fact, significantly greater than the lower bound given in Corollary 2, which was $0.15n$. Now let $2^6 \leqslant n \leqslant 2^{12}$, $\kappa = \frac{n}{2^6}$ and privacy parameters stays the same. See Figure 2.4. Again we can see that the error magnitude is unacceptably high, greater than $0.2n$. Note that the result itself can be at most n. See that even if we ignore and throw away all the data and decide about the value for each user via coin toss, it yields an expected error of at most $0.5n$ so the same order of magnitude as the Binary Protocol. Moreover, the noise is independent from the data, so such error could be very problematic, especially if the sum of the real data is small (e.g $o(n)$). In such case the noise could be greater than the data itself. We can also see how great the errors will be for constant value of $\kappa = 5$. See Figure 2.5.

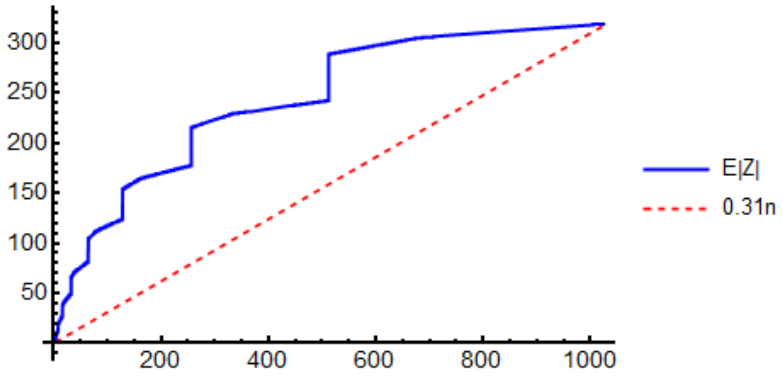


Figure 2.3: Error in Binary Protocol with $\epsilon = 0.5$, $\delta = 0.05$ and $\kappa = \lfloor \log_2(n) \rfloor$.

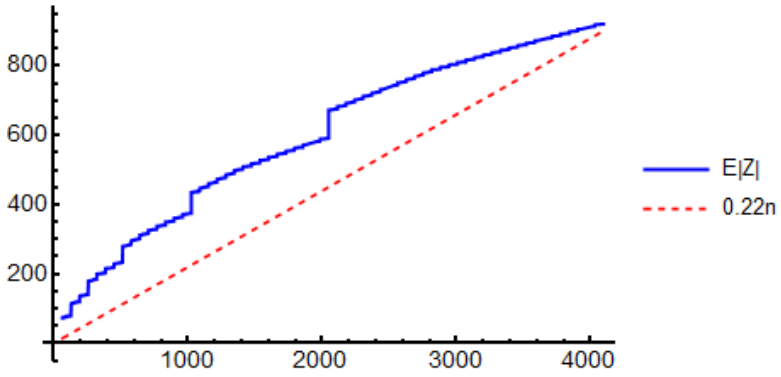


Figure 2.4: Error in Binary Protocol with $\epsilon = 0.5$, $\delta = 0.05$ and $\kappa = \left\lfloor \frac{n}{2^6} \right\rfloor$.

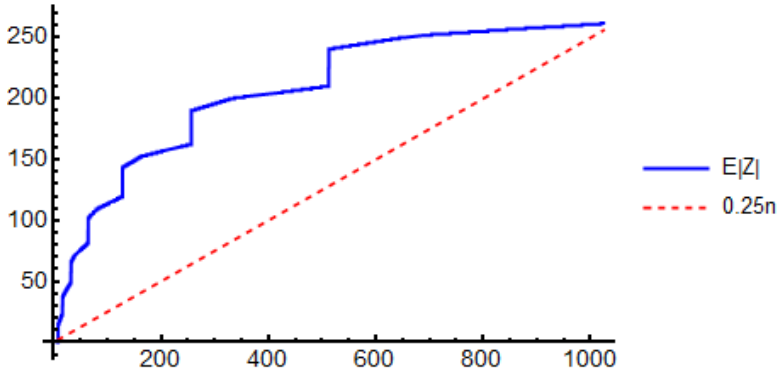


Figure 2.5: Error in Binary Protocol with $\epsilon = 0.5$, $\delta = 0.05$ and $\kappa = 5$.

2.3 Precise Aggregation Algorithm with Local Communication

In this part we present an alternative protocol PAALC (Precise Aggregation Algorithm with Local Communication) that in some scenarios offers much better accuracy of aggregated data when failures occur, while preserving privacy. In fact, our protocol works in a substantially different way and for slightly modified model. Thus, despite its performance and accuracy outperforms the original protocol of Chan et al., they are not fully comparable.

First of all, we assume that users may communicate. Let us stress that the communication is limited to a small circle of "neighbors". The idea behind the presented construction is to take advantage of some natural structures emerging

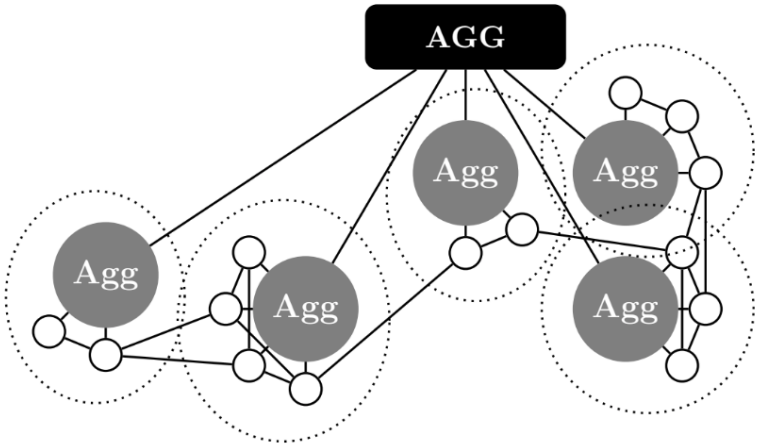


Figure 2.6: Example of a clusterized network with global aggregator ($\mathbf{AGG}$) and local aggregators ($\mathbf{Agg}$) marked.

in distributed systems. Apart from logical connections between each user and a server/aggregator there are also some direct links between individual users, be it either secure channel of communication or trust relation. Clearly, such model is not adequate for some real-life problems discussed in [17], for example in sensor fields with unidirectional communication. Thus there are applications where the original approach without any local communication is the only one possible.

2.3.1 Modified Model

We assume that the network consists of n users where $V = \{v_1, v_2, \ldots, v_n\}$ denotes the set of all users (or their respective vertices) as well as the aggregator $\mathbf{AGG}$ and a set of $k < n$ *local aggregators* $\mathbf{Agg}_1, \ldots, \mathbf{Agg}_k$. Please note that the local aggregators may be separate entities but without any significant changes they may be selected from the set of regular users V. The only issue with this approach is that we have to ensure that the local aggregator is either selected during the aggregation round or it cannot fail during **a single** execution of aggregation process. We assume that each user is assigned to **exactly one** local aggregator. We denote the set of nodes assigned to the local aggregator $\mathbf{Agg}_i$ by V_i. An example of the network's topology is depicted in Figure 2.6.

We can derive a graph $G = (V, E)$ from the network structure, where V are all the nodes and the set of edges is created based on the ability to establish com-

munication (e.g., transmission range in a sensor network, friendship relation in a social network). Namely, $\{v, v'\}$ is an edge if and only if v and v' are *neighbors* and can communicate via secure private channel. In our protocol we assume that each node can perform some basic cryptographic operations and has access to an independent source of randomness. By $N(v)$ we denote a set of such vertices v' of G that there exists an edge $\{v, v'\}$. Security of the protocol described in Section 2.3.3 depends on the structure of graph G, and how many parties the adversary can corrupt. Discussion on security of the protocol can be found in Section 2.4.

Adversary The Adversary may corrupt and therefore control a subsets of users, local aggregators and the aggregator. He can read all messages the controlled parties sent or received. Nodes controlled by the Adversary may decide not to add the noise, even if they should according to the protocol. Note, however, that sabotaging the whole protocol (e.g. by making the result incorrect or corrupted in any way) is not the goal of the Adversary. The goal of the Adversary in this model is to obtain sum of aggregated data for any subset of uncorrupted users with worse privacy parameters than those guaranteed. If the Adversary cannot obtain such information, we consider the protocol differential privacy preserving with appropriate parameters.

2.3.2 Building Blocks

For obtaining high level of data privacy we combine cryptographic techniques with data perturbation methods typical for research concentrated on differential privacy of databases. First we recall the Decisional Diffie-Hellman assumption.

Definition 21 (Decisional Diffie-Hellman assumption)**.** Consider a cyclic group $\mathbf{G}$ of order q. Given (g, g^a, g^b, g^c) for a randomly chosen generator $g \in \mathbf{G}$ and random $a, b, c \in 0, \ldots, q-1$ for the adversary g^{ab} and g^c are computationally indistinguishable.

We say that Decisional Diffie-Hellman problem is hard in group $\mathbf{G}$ if the group satisfies the Decisional Diffie-Hellman assumption.

The first technique we use in our protocol is a homomorphic encryption scheme based on original ElGamal construction enriched by some extra techniques introduced in [34]. More precisely, encrypted messages can be aggregated and reencrypted. Moreover one can add an extra encryption layer to a given ciphertext, in such way that the message can be decrypted only using both respective keys.

Let p denote a large prime number and let $\mathbf{G}$ be a group of order p such that the Decisional Diffie-Hellman problem is hard. Let g be a generator of $\mathbf{G}$. Let $\mathrm{sk}, \mathrm{sk}'$ be some private keys and $g^{\mathrm{sk}}, g^{\mathrm{sk}'}$ are respective public keys. For the sake of clarity we skip some technical details (i.e., choice of the group size, generators etc.) as well as full security discussion of this encryption scheme. These techniques can be found for example in [34].

Encryption of '1'

A pair $\mathrm{Enc}_{\mathrm{sk}}(1) = (g^r,\, g^{r\cdot \mathrm{sk}})$ for a random $r \in \mathbf{Z_p}$ is an encryption of 1 using secret key sk.

Re-encryption

A ciphertext representing 1 can be re-encrypted. Namely, one can get another ciphertext representing '1', **without private key**. Namely having $\mathrm{Enc}_{\mathrm{sk}}(1) = (g^r,\, g^{r\cdot \mathrm{sk}})$ one can choose r' and compute $\mathrm{Re}(\mathrm{Enc}_{\mathrm{sk}}(1)) = (g^{r\cdot r'},\, g^{r\cdot r'\cdot \mathrm{sk}})$ that represents 1 as well.

Adding layer of encryption

Having a ciphertext $\mathrm{Enc}_{\mathrm{sk}}(1) = (g^r,\, g^{r\cdot \mathrm{sk}})$ a party having private key sk' can add encryption layer to a ciphertext obtaining

$$\mathrm{Enc}_{\mathrm{sk}+\mathrm{sk}'}(1) = ((g^r)^{r'},\, (g^{r\cdot \mathrm{sk}})^{r'} \cdot (g^r)^{r'\mathrm{sk}'}) = (g^{r\cdot r'},\, g^{r\cdot r'\cdot(\mathrm{sk}+\mathrm{sk}')}).$$

Filling the ciphertext

Having $\mathrm{Enc}_{\mathrm{sk}}(1) = (g^r,\, g^{r\cdot \mathrm{sk}})$ for any message $m \in G$ one can compute

$$\mathrm{Enc}_{\mathrm{sk}+\mathrm{sk}'}(\mathrm{m}) = (g^r,\, g^{r\cdot \mathrm{sk}} \cdot m).$$

Partial decryption

Having $\mathrm{Enc}_{\mathrm{sk}+\mathrm{sk}'}(\mathrm{m}) = (g^{r\cdot r'},\, g^{r\cdot r'\cdot(\mathrm{sk}+\mathrm{sk}')}m)$ and a private key sk' for $m \in G$ one can remove one layer of encryption and obtain

$$\mathrm{Enc}_{\mathrm{sk}}(\mathrm{m}) = \left(g^{r\cdot r'},\, \frac{g^{r\cdot r'\cdot(\mathrm{sk}+\mathrm{sk}')}m}{(g^{r\cdot r'})^{\mathrm{sk}'}}\right) = (g^{r\cdot r'},\, g^{r\cdot r'\cdot \mathrm{sk}}m).$$

2.3.3 Protocol Description

During the protocol, we assume that the aggregator $\mathbf{AGG}$ has a private key sk, moreover each of the local aggregators $\mathbf{Agg}_i$ has its own private key sk_i. We also assume that there is a public parameter g, that is a generator of some finite group $\mathbf{G}$, in which Decisional Diffie-Hellman problem is hard. By $\mathrm{Enc}_{\mathrm{sk}}(\mathrm{c})$ we denote the encryption structure introduced previously in Section 2.3.2. Let us assume that each user v has a private value ξ_v from the range $[0, \Delta]$. Furthermore, we assume that there are private channels between some of the users (underlying communication graph). The final aim is to provide $\mathbf{AGG}$ the sum $\sum_{v \in V} \xi_v$ perturbed in such a way that the privacy (expressed in terms of differential privacy) of all $v \in V$ is preserved. Clearly, the privacy of users can be endangered both by revealing the output as well as by collecting information about the aggregation process. The description of the protocol is presented below.

Setup

- $\mathbf{AGG}$ broadcasts to the local aggregators $\mathrm{Enc}_{\mathrm{sk}}(1)$.
- Each of the local aggregators $\mathbf{Agg}_i$ constructs $\mathrm{Enc}_{\mathrm{sk}+\mathrm{sk_i}}(1)$ and publishes it for all users from V_i.

The setup phase is performed only once during network's lifetime.

Aggregation for node v

- For each node $v' \in N(v)$ generate a random value $x_{v'}^{v} \in \mathbf{G}$.
- Using a private channel send each value $x_{v'}^{v}$ to the appropriate neighbor v'.
- Having received all $x_{v}^{v'}$ from each of the neighbors, select random r_v from $Geom^{\beta}(\alpha)$ and calculate

$$c_v = \sum_{v' \in N(v)} x_{v}^{v'} - \sum_{v' \in N(v)} x_{v'}^{v} + r_v + \xi_v.$$

- Compute $\mathrm{Re}(\mathrm{Enc}_{\mathrm{sk}+\mathrm{sk_i}}(\mathrm{g}^{\mathrm{c_v}}))$ and send it to $\mathbf{Agg}_i$.

Aggregation for local aggregator $\mathbf{Agg}_i$

- Having received $\mathrm{Enc}_{\mathrm{sk}+\mathrm{sk_i}}\left(\mathrm{g}^{\mathrm{c_v}}\right)$ from all nodes from V_i, compute

$$\mathrm{Enc}_{\mathrm{sk}}\left(\mathrm{g}^{\mathrm{c_v}}\right) = \left(g^{r_i},\, \frac{g^{r_i(\mathrm{sk}+\mathrm{sk}_i)+c_v}}{g^{r_i\cdot \mathrm{sk}_i}}\right).$$

  This operations result in obtaining *shares*:

  $\mathrm{Enc}_{\mathrm{sk}}\left(\mathrm{g}^{\mathrm{c_{v_1}}}\right) = (g^{r_{v_1}}, g^{r_{v_1}\cdot \mathrm{sk}+c_{v_1}}),\ \ldots,\ \mathrm{Enc}_{\mathrm{sk}}\left(\mathrm{g}^{\mathrm{c_{v_l}}}\right) = (g^{r_{v_l}}, g^{r_{v_l}\cdot \mathrm{sk}+c_{v_l}}),$

  of all $l = |V_i|$ users from $|V_i|$.

- Compute

$$\begin{aligned}\mathrm{Enc}_{\mathrm{sk}}\left(\mathrm{g}^{\mathrm{c_{v_1}}+\cdots+\mathrm{c_{v_l}}}\right) &= \left(\prod_{i=1}^{l} g^{r_i},\, \prod_{i=1}^{l} g^{r_i \mathrm{sk}+c_{v_i}}\right)\\ &= \left(g^{\sum_{i=1}^{l} r_i},\, g^{(\sum_{i=1}^{l} r_i)\mathrm{sk}+\sum_{i=1}^{l} c_{v_i}}\right).\end{aligned}$$

- Send the value $\mathrm{Enc}_{\mathrm{sk}}\left(\mathrm{g}^{\mathrm{c_{v_1}}+\cdots+\mathrm{c_{v_l}}}\right)$ to the aggregator $\mathbf{AGG}$.

Final aggregation

- Having received the aggregated values from each V_i, for each of those values $\mathbf{AGG}$ calculates $y_i = g^{\sum_{v\in V_i} c_v}$, using its private key sk for each $i = 1, \ldots, k$. Then computes

$$y = \prod_{i}^{k} y_i = \prod_{i} g^{\sum_{v\in V_i} c_v} = g^{\sum_{v\in V} c_v}.$$

- Then $\mathbf{AGG}$ computes discrete logarithm of y as a final (perturbed) value being a sum of all $\sum_{v\in V} \xi_v$.

Similarly to previous papers (including [17, 62]), we utilize the following method: if we know that each user $v \in V$ has a value from an interval of moderate size $\xi_v \in [0, \Delta]$ then the sum of values of all ξ_v's cannot exceed $n\Delta$. Thus one

can find a discrete logarithm for $g^{\sum_{v\in V}\xi_v}$ even if finding a discrete logarithm of g^r is not feasible for a random element $r \in \mathbf{G}$. Using Pollard's Rho method this can be completed in average time $O(\sqrt{n\Delta})$. An example of node's communication is shown in Figure 2.7. Note that the protocol depends on two security parameters β and α. They strongly depend on the topology of the underlying graph. We discuss this issue in Section 2.4.

2.4 Analysis of PAALC

In this section we outline the analysis of the presented aggregation protocol with respect to correctness, level of privacy provided and error of the result obtained by the aggregator. The analysis is slightly more complicated since the parameters of the protocol strongly depend on the underlying network. However, we argue that they offer very good properties for wide class of networks. We prove that the proposed protocol guarantees very good accuracy even facing a massive failures and compromising of nodes. Half of nodes may fail or cooperate with the adversary (in fact this result can be generalized to any constant fraction of users). One can instantly observe that the analysis can be extended for smaller δ for the price of moderate increasing of the expected noise.

Note that if a graph guarantees a specific level of privacy then more dense graph (with some added edges) offers at least the same level of privacy. Thus it is enough if each user adds some "randomly" chosen neighbors to protect the privacy in any network. We elaborate more about this in Chapter 3, which is devoted to such graph enrichments.

Correctness First, let us look at the result obtained by the aggregator $\mathbf{AGG}$ in the last step of the protocol. This is a discrete logarithm of $g^{\sum_{v\in V} c_v}$. Let us observe that

$$\begin{aligned}\sum_{v\in V} c_v &= \sum_{v\in V}\left(\sum_{v'\in N(v)} x_v^{v'} - \sum_{v'\in N(v)} x_{v'}^{v} + r_v + \xi_v\right)\\ &= \sum_{v\in V}\sum_{v'\in N(v)} x_v^{v'} - \sum_{v\in V}\sum_{v'\in N(v)} x_{v'}^{v} + \sum_{v\in V}\xi_v + \sum_{v\in V} r_v = \sum_{v\in V}\xi_v + \sum_{v\in V} r_v.\end{aligned}$$

The value $\sum_{v\in V} c_v$ is the exact sum of values kept by nodes $\left(\sum_{v\in V}\xi_v\right)$ and sum of all the noises $\left(\sum_{v\in V} r_v\right)$. This leads to two conclusions. First, the result is

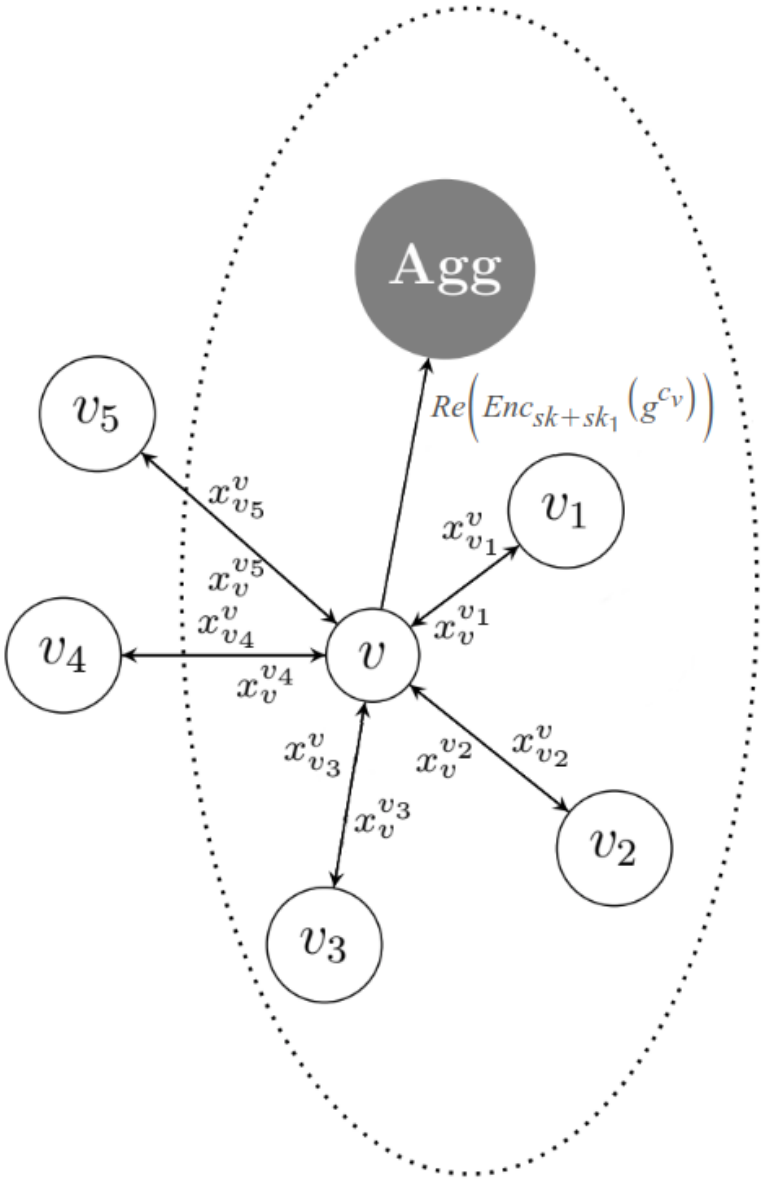


Figure 2.7: An example of communication in a single aggregation round for node v. The dotted line marks the set of nodes assigned to a single local aggregator $\mathbf{Agg}$. Note that neighbors may have different local aggregators.

correct. Second, retrieving the data using Pollard's Rho method (or even brute force method) is feasible since the absolute value of the first sum has to be smaller than $n\Delta$. One can easily see that the sum of added noises is of the magnitude $O(n)$ with high probability.

Privacy protection We assume that the encryption scheme $\mathrm{Enc}_{\mathrm{sk}}(\cdot)$ is *semantically secure*. In particular, in our protocol, the local aggregator $\mathbf{AGG}_i$ cannot learn the contributions sent to $\mathbf{AGG}_j$ for $i \neq j$ without access to keys sk_j and sk.

Note that all neighboring users exchange a purely random values $x_v^{v'}$'s that finally cancel-out, however as long as they remain unknown to the adversary, they perfectly obfuscate the results sent to the aggregator.

We recall the following fact.

Fact 6. *(From [17]) Let $\epsilon > 0$. Let u, v be integers such that $|u - v| \leq \Delta$ for fixed $\Delta \in \mathbb{N}^+$. Let R be a random variable having distribution $Geom(\exp(\frac{\epsilon}{\Delta}))$. Then for any integer k*

$$\mathbb{P}[v + R = k] \leq \exp(\epsilon)\mathbb{P}[u + R = k].$$

Moreover, we will call those users that are not being corrupted by the Adversary or prone any kind of failure *uncompromised* users. Now we can state and prove

Theorem 6. *Let us assume that PAALC with parameter $\alpha = \exp(\frac{\epsilon}{\Delta})$ is executed in the network represented by a graph $\mathcal{G} = (V, E)$ and $\mathcal{G}'$ is a subgraph of $\mathcal{G}$ induced by the set of uncompromised users V^H. Moreover we assume that each user v contributes a value $\xi_v \in [0, \Delta]$.*

If in each connected component $\mathcal{S}$ of $\mathcal{G}'$ there is a user s, such that its added noise r is taken from $Geom(\exp(\frac{\epsilon}{\Delta}))$, then PAALC preserves computational $(\varepsilon, 0)$-differential privacy.

Proof Let $\Xi = \sum_{s \in S} \xi_s$ and let Ξ' be the same sum with a changed single value ξ_s. By the assumption about the range of the aggregated values we get $|\Xi' - \Xi| \leq \Delta$. Let r be a random variable taken from the symmetric geometric distribution $Geom(\exp(\frac{\epsilon}{\Delta}))$. From Fact 6 we know that $\mathbb{P}[\Xi + r = k]$ may differ from $\mathbb{P}[\Xi' + r = k]$ by at most a multiplicative factor $\exp(\epsilon)$. However, because the encryption scheme is semantically secure, we know that the adversary may learn nothing more than the sum of all values from the component $\mathcal{S}$. □

From this theorem follows next corollary.

Corollary 3. *If PAALC is executed on a graph such that a subgraph induced by the set of uncompromised users V^H is connected and with probability at least $1-\delta$ at least one uncompromised user adds its value r from $Geom(\exp(\frac{\epsilon}{\Delta}))$ then PAALC computationally preserves (ε, δ)-differential privacy.*

Translating into real terms Theorem 6 with Corollary 3 mean that if the connections between honest users are dense enough and we can somehow guarantee that at least one honest node adds the noise, the system is secure. The core of the problem is to judge if a real-world networks are dense enough and what parameters of adding noise are sufficient.

Accuracy The level of accuracy and security in this protocol strongly depends on the graph topology and chosen security parameters. We consider a $G(n,p)$ graph, for fixed p where the Adversary controls up to $n-m$ randomly chosen users.

Theorem 7. *Let us consider a random network with n nodes. Each of possible $\binom{n}{2}$ connections (edges) is independently added to the network with probability $p \geq \frac{8\log n}{n}$. Let $\mathcal{S}$ be a subgraph induced by a subset of at least $m \geq n/2$ randomly chosen nodes. Then $\mathcal{S}$ is connected with probability at least $1-1/n$.*

Proof Let us note that $\mathcal{S}$ is **not** connected if and only if there exists a subset of nodes from $\mathcal{S}$ with cardinality $1 \leq k \leq m/2$ such that there is no connection to any of the remaining $m-k$ nodes. For a given subset of $\mathcal{S}$ of cardinality k probability that no edge connects it to other $m-k$ nodes of $\mathcal{S}$ is $(1-p)^{k(m-k)}$.

Let A_k be an event that there exists such a "cut-off" subset of cardinality k. Using union bound argument we get

$$\mathbb{P}[A_k] \leq (1-p)^{k(m-k)} \binom{m}{k}.$$

Probability that $\mathcal{S}$ is not connected is equivalent to the event $A_1 \cup \ldots \cup A_{m/2}$. Again, using union bound

$$\mathbb{P}[A_1 \cup \ldots \cup A_{\frac{m}{2}}] \leq \sum_{k=1}^{m/2} \mathbb{P}[A_k] \leq \sum_{k=1}^{m/2} (1-p)^{k(m-k)} \binom{m}{k} \leq$$
$$\leq \sum_{k=1}^{m/2} (1-p)^{k\frac{m}{2}} \binom{m}{k} = (\star).$$

Since $\binom{m}{k} \leqslant m^k$ we get

$$(\star) \leq \sum_{k=1}^{m/2} \left((1-p)^{\frac{m}{2}} m\right)^k \leq \sum_{k=1}^{\infty} \left((1-p)^{\frac{m}{2}} m\right)^k = \frac{(1-p)^{m/2} m}{1-(1-p)^{m/2} m} = (\star\star).$$

Since the function $f(x) = \frac{a^x x}{1-a^x x}$ is decreasing for $x > -\frac{1}{\log(a)}$ (if $0 < a < 1$) and from the assumption that $m \geq n/2$ we have

$$(\star\star) \leq \frac{(1-p)^{n/4} \frac{n}{2}}{1-(1-p)^{n/2} \frac{n}{2}} .$$

Applying inequality $\exp(x) \geq 1 + x$ and substituting $p = \frac{8 \log n}{n}$ we obtain

$$(\star\star) \leq \frac{\exp\left(-\frac{8\log(n)}{n}\right) \frac{n}{2}}{1-1/2} \leq \exp\left(-\log(n^2)\right) n = \frac{1}{n},$$

which concludes the proof of this theorem.

□

From Theorem 7 we learn that a "typical" network of n nodes with random connections such that the average number of neighbors is $8 \log n = \Theta(\log n)$ is dense enough even if the adversary is able to compromise as much as $n/2$ nodes.

If we have guaranteed at least $n/2$ honest (uncompromised and working) nodes one may note that the probability that none of them adds the noise is at least $(1-\beta)^{n/2}$. To have $(1-\beta)^{n/2} \leq \delta$ one needs to have β such that $\log(1-\beta) \leq \frac{2\log\delta}{n}$. Since $\log(1+x) \leq x$ for $x > -1$ it is enough to use $\beta \geq \frac{2\log(1/\delta)}{n}$. Clearly the expected error cannot exceed $2\sqrt{\log(1/\delta)}$ for $\beta = \frac{2\log(1/\delta)}{n}$.

2.5 PAALC and Binary Protocol Comparison

In this section we experimentally compare the Binary Protocol from [17] and our PAALC described in Section 2.3. We conduct an experiment on real data from Facebook social network collected in SNAP dataset by Stanford University (see [44] and [49]), where nodes denote users and edges denote friend relation. We have 4039 nodes and assume that each user holds one bit of information, i.e. value $x_i \in \{0, 1\}$. For number of node failures $\kappa \in \{0, 1, \ldots, 200\}$ we check what is the error size in our protocol with parameters $\epsilon = 0.5$ and $\delta = 0.05$. Then we compare it to the Binary Protocol with the same privacy parameters.

Firstly, in Figure 2.8 we can see what is the average fraction of nodes remaining in the giant component after κ node failures. One can easily see, that the overwhelming majority of nodes remain connected in a single, giant component. Our protocol will preserve the privacy of these nodes. We emphasize, that the nodes remaining out of the giant component may be prone to privacy loss. Probability of being out of the giant component may be also added to the δ parameter.

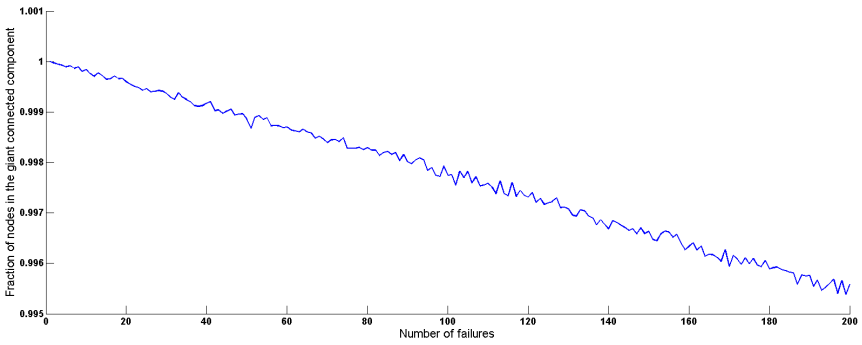


Figure 2.8: The average fraction of nodes remaining in the giant component after κ failures.

Observe that the Binary Protocol does not utilize the connections and communication between users in any way. Our protocol, on the other hand, depends on the structure of the underlying graph, which means that on more dense dataset it will perform better than on sparse graph. In Figure 2.9 one can see what is the size of error in both protocols.

See that the additional error in PAALC is constant, while in the Binary Protocol the errors are much higher with high probability. Recall that the real sum of the data is at most 4039 (if all users hold 1). Thus, error of size of magnitude

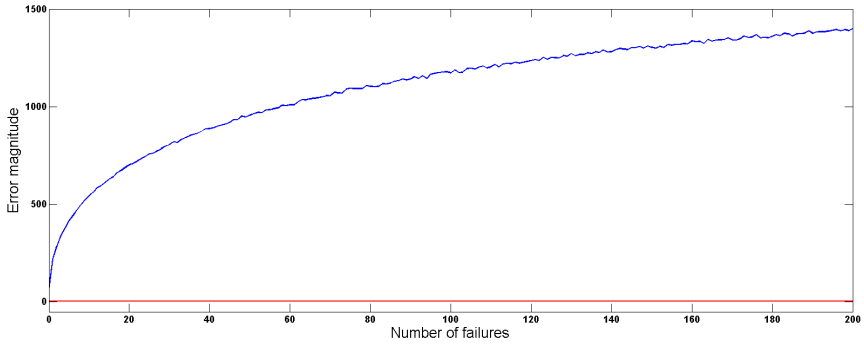


Figure 2.9: Blue line denotes the error in the Binary Protocol, red line denotes the error in PAALC

10^3 renders the aggregated data not suitable for statistic inference. On the other hand, our protocol gives constant error of size approximately 5, which makes the aggregated data not only private, but also useful for statistical analysis. Unfortunately, our vast error decrease comes at a price of not protecting the privacy of these nodes which do not belong to the giant component. This, obviously, is a significant drawback of our protocol. However, it can be mitigated by, for example enriching the graph with additional edges or doing an additional check whether specific node belongs to the giant component or not (then the outlying nodes would have to always add noise).

We again want to emphasize that PAALC and the Binary Protocol are not fully comparable. The Binary Protocol gives the privacy guarantee to all users and does not require communication between them. However, it is not robust to failures, despite the fact that it is designed precisely as a fault tolerant protocol. In particular, even if less than 5% users are prone to failure, the error in the aggregated data is too big for the data to remain useful for any reasonable statistical purposes. On the other hand, PAALC requires some communication (albeit very limited) between users and, maybe more importantly, strongly depend on the connections between users (communication channels). The denser the network, the more secure PAALC is. Without the improvements which we mentioned in previous paragraph, the privacy of users not belonging to the giant connected component in PAALC is prone to attacks.

To conclude, in practical setting where we have quite dense network and we expect more than just a limited, constant number of failures but rather a few percent of failures amongst users, the Binary Protocol returns the aggregated data

with such a huge error, that it might not be appropriate for any reasonable statistical analysis. For such scenarios, if we can pay the price of trust and communication between some of the nodes, PAALC gives us constant size error, significantly smaller than the Binary Protocol.

Chapter 3

Amplification of Privacy Using Local Knowledge in Faulty Network

In naturally emerging networks users usually have a limited number of (semi)-trusted contacts. Clearly, they also have mostly local knowledge about the whole network. We consider a distributed system that consists of nodes which need to constitute a huge, connected group in an efficient way and without knowledge of global network topology.

It has been noticed that to perform some operations (e.g. data aggregation) in distributed systems, it is often necessary to involve a large number of users. It is worth mentioning that for real-life applications one has to take into account that either random failures in the network or an external Adversary could cause some of the nodes to malfunction. This can lead to the network being disconnected and some users, or groups of users, isolated.

In this chapter we propose and investigate **local** strategies for constructing large groups of users based only on local relations of trust with surprisingly low communication and computation overhead. Moreover, these strategies are effective even facing a powerful adversary capable of controlling a vast **majority** of users. This is non-trivial property in real-life networks, as those are usually modeled using preferential attachment graphs, which are extremely prone to attacks on the hub nodes. We show that using our protocols we can achieve similar robustness as Erdős-Renyí graphs, which, on the contrary, are very resistant against attacks focused on chosen nodes.

We provide comprehensive tests on datasets representing **real-life** networks for our protocols. Moreover, we prove some properties of these networks while formally assuming that they are generated as a *preferential attachment process*.

There is a vast research showing that complex, real life networks exhibit a structure that can be modeled by preferential attachment graphs (see for example [2, 6, 64]). We present the preferential attachment model in Subsection 1.4.2.

We believe that our results can be also seen as a contribution to fundamental observation about the nature of real-life networks. These results may help to design protocols, whenever it is necessary to gather a big group of users in highly dynamic or even adversarial settings. From mathematical point of view our contribution can be seen as a problem of constructing strategies for strengthening connectivity of a random graph by adding (locally and independently) some number of extra edges.

3.1 Model

In this section we introduce a formal model and present definitions and notation that will be used throughout the chapter.

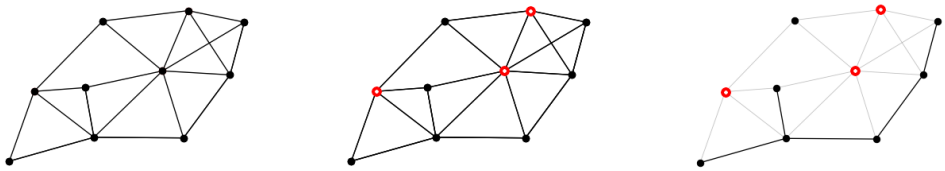
In a real-life social network there are usually a few well-known and somewhat trusted parties in the whole community. We model them by *fat nodes*.

Definition 22 (Set of fat nodes, F)**.** Let $G(V, E)$ be a graph such that $|V| = n$. By $F \subset V$ we denote a subset of vertices whose degrees vary from $\frac{an}{\log n}$ to $\frac{bn}{\log n}$ for some fixed constants a, b. We call them fat nodes.

Existence of such fat nodes is a phenomenon typical for structures governed by preferential attachment model and there are relatively few of these nodes (typically up to $\log n$).

Now, we want to model a situation when some of the nodes in graph may either be offline due to some random failure or even specifically targeted and destroyed by an external Adversary. We say that a node is *healthy* if it is neither corrupted by the Adversary nor offline for any reason. In other words, *healthy* node can correctly perform given protocol without revealing any information to the Adversary.

If the vertex is not *healthy* we call it *unhealthy*. We consider all unhealthy vertices as removed from the graph together with their incident edges. At the same time we want to prevent healthy nodes from being isolated from the rest of the network (we aim to have a group of healthy nodes which could constitute a huge, connected group to be able to perform various security-enhancing protocols on the network). We propose the following notion of graph robustness in distributed systems.



(a) Initial graph (b) Corrupted nodes (c) Graph after attack

Figure 3.1: Adversarial attack example

Definition 23. We say that a graph is ξ-*strong*, for $0 \leqslant \xi \leqslant 1$, if a subgraph induced by its healthy nodes has largest connected component of size at least ξn, where n is the number of healthy nodes.

Example 3. In Figure 3.1 we show an intuition behind both healthy nodes and ξ-strength of a graph. In the first figure we can see a fully connected network with all nodes healthy. In the second figure the Adversary has chosen 3 nodes (marked by red color) to corrupt. Then these nodes and incident edges are removed from the graph and we are left with 7 nodes out of which 6 are connected. It means that after the attack ξ-strength of the remaining graph is $\frac{6}{7}$.

If the graph is ξ-strong, it means that there exists a connected structure, that is **not** controlled by the Adversary, containing at least ξn out of n nodes. In privacy settings, this allows to provide a common response secured in such way that the Adversary cannot observe separate inputs of nodes but the aggregated value of a large set of nodes.

Obviously the most desired situation is when the graph is 1-strong, which means that all healthy nodes are connected. Clearly, corruption of a significant number of nodes can dramatically decrease the ξ-strength. To mitigate the attack we enrich the graph by adding some edges between users. For practical reasons all these operations need to be simple (computationally affordable for each node) and **local**, namely each node does not have knowledge about the whole network, it only knows list of its neighbors and, potentially, several fat nodes).

We have a network with underlying graph $G = (V, E)$. We define *Disconnection Game* with the Adversary $\mathcal{A}$ and protocol $\mathcal{P}$, denoted by $\mathcal{DG}(G, \mathcal{A}, \mathcal{P})$ in the following way:

1. The set of edges E is enriched by adding edges chosen between pairs of unconnected nodes according to protocol $\mathcal{P}$. Rules of adding edges depend

on specific game instantiation. This resulting graph is $G_P = (V, E \cup E_P)$, where E_P is the set of edges added after $\mathcal{P}$ was applied.

2. The Adversary chooses, according to restrictions in this game instantiation (e.g. is given no information or all the degrees of nodes), a subset C of nodes. The nodes belonging to C are *corrupted* and removed from the graph with their incident edges denoted by E_C. Note that the Adversary knows only the initial graph G so C does not depend on the set E_P. This reflects the assumption that the Adversary does not know the choices and connections added during protocol $\mathcal{P}$ between healthy nodes. The resulting graph is $G_A = (V \setminus C, (E \cup E_P) \setminus E_C)$.

The outcome of the game is the fraction of nodes belonging to the biggest connected component in graph G_A or, in other words, ξ-strength of graph G_A.

The game presented in this section is connected with the problem of robustness of the network (see for example [72, 73]). It is, however, worth mentioning that unlike mentioned papers, we require that the enhancing protocol is done in a distributed way and without knowledge of global topology of the graph. Moreover, we pick rather strong notion of robustness, namely the size of the largest connected component.

We assume that the corruption of nodes can either be done in a random way (Random Failures strategy) or the Adversary can choose to attack nodes with highest degrees (Targeted Attack strategy).

- *Random Failures* - the Adversary chooses subset C of nodes using uniform distribution. In other words, the subset is chosen uniformly at random out of all possible subsets of a specific size.

- *Targeted Attack* - the Adversary chooses subset C of nodes with highest degrees. In other words, the set of nodes is sorted decreasingly by its degree and then the Adversary picks m first nodes as subset C. One can easily see that this strategy is far more destructive for the structure of remaining, healthy nodes.

3.2 Security Enhancing Protocols

In this section we present three local protocols enriching the set of relations between users of the network which improve ξ-strength of its structure. We prove

their properties both in analytic (Section 3.3) and experimental (Section 3.4) way for underlying graphs typical for social networks.

3.2.1 k-Two Steps Friend Finder Algorithm

We want to leverage the local knowledge that each node has, namely the list of its neighbors. The idea is that by asking our neighbor to introduce us to his neighbor (one can think of it as connecting to a "friend of a friend") an arbitrary node can improve the chance of being in the big, connected component of a graph. Such protocol can be performed without any additional knowledge and in a completely distributed way.

The node which wants to improve its chances of being in the big component asks its friend (chosen uniformly at random from list of its neighbours) to recommend it yet to another friend. Namely, our new friend is a former "friend of a friend" that is temporarily added to the list of connections as a separated contact used for privacy-preserving actions. Note that due to computational or communication cost, some nodes might not want to actively participate in the protocol. Still, however, they could be used as "friend of a friend" and therefore their chance to remain in the largest connected component would increase. This procedure is iterated k times, namely each participating node asks k randomly chosen friends for recommendations. That would result in obtaining (at most) k new friends. Note that sometimes it might happen that a specific "friend of a friend" will be recommended more than once.

Formally speaking, every node that wants to actively participate in the protocol performs (k times) a random walk of length two starting from himself. Note that one could propose different length of the random walk, our choice of length is to minimize communication and keep the protocol as local as possible.

Definition of the k-Two Steps Friend Finder (k-2SFF, for short) is presented as Algorithm 1. This is a procedure for each node v, initially $m = k$ if node is participating or $m = 0$ otherwise. Moreover, N denotes the array consisting of the neighbors of node v. We will also denote $N[i]$ as i-th element of an array. By *rand*(*a, b*) we denote a function that returns integer chosen uniformly at random from $\{a, a+1, \ldots, b-1\}$. Here we also briefly describe messages used throughout the protocol

- REQ<v> - message requesting for a friend recommendation for node v,
- FRD<v> - message consisting of a friend (node v) recommendation,

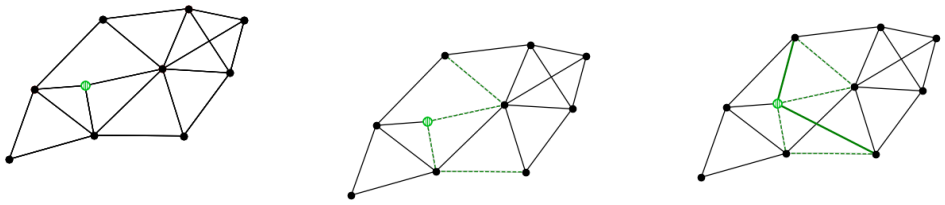
- EDG<v> - message consisting of a connection proposition from node v,
- ACK<v> - message acknowledging a connection proposition to node v.

```
1  foreach node v do
2      while true do
3          if received msg = REQ <u> then
4              1. r ← rand(0, |N[v]|).
5              2. Send message "FRD <N[r]>" to node u.
6          else if received msg = FRD <u> then
7              1. Send message "EDG <v>" to node u.
8          else if received msg = EDG <u> then
9              1. Add u to list of neighbors.
10             2. Send message "ACK <v>" to node u.
11         else if received msg = ACK <u> then
12             1. Add u to list of neighbors.
13             2. m ← m - 1.
14         else if m > 0 then
15             1. r ← rand(0, |N[v]|).
16             2. Send message "REQ <v>" to node N[r].
```

Algorithm 1: k-2SFF

Note that k-2SFF can be performed by a node without any global knowledge of the underlying graph, except its neighbors. Moreover, it can be done in a fully distributed manner, with $O(kn')$ messages sent in the network, where $n' \leq n$ is the number of nodes participating in the protocol.

Example 4. In Figure 3.2 we show an example for 2-2SFF. In Figure 3.2a one can see the initial graph. In the next picture one node, marked by green color, is performing the 2-2SFF protocol and creates two friend connections, also marked by green color. Then, in the rightmost picture we see graph with two added edges. Now see in Figure 3.3 how the attack would look like. First we see initial, but already enriched graph, then the Adversary chooses nodes to corrupt (marked by red color, same ones as in Figure 3.1 in previous section). See that this time, the remaining structure is connected, there are no isolated nodes. Finally one can see in Figure 3.4 the outcome after the attack. In Figure 3.4a we see the result in case where one node used 2-2SFF, while in Figure 3.4b we see the case

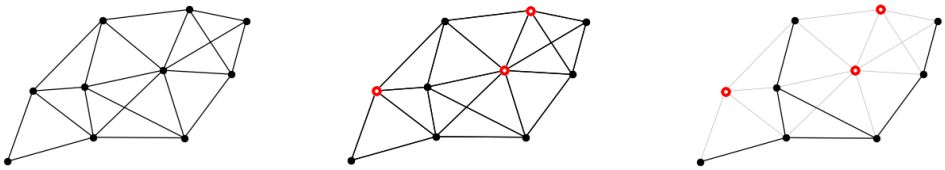


(a) Initial graph, with a single node starting 2-2SFF procedure (green)

(b) Querying neighbors

(c) Newly established edges addition

Figure 3.2: Example for 2-2SFF protocol.



(a) Enriched graph

(b) Corrupted nodes

(c) Graph after attack

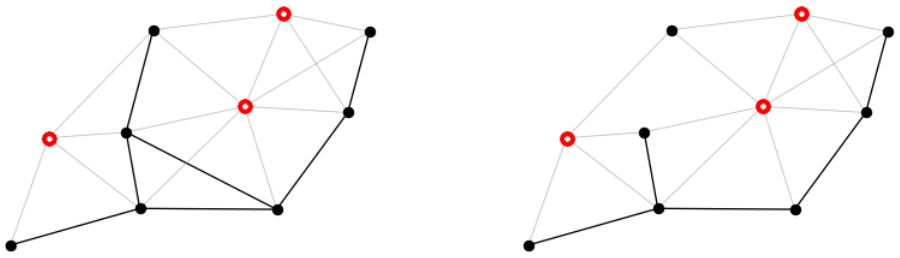
Figure 3.3: Adversarial attack example after 2-2SFF protocol has been applied.

where no protocol was applied. This example clearly shows that our protocol helps preventing the nodes from being isolated after the attack.

3.2.2 k-Ask Fat For a Friend Algorithm

The approach in this protocol is substantially different to previous one. Here we want to rely on the preferential attachment properties of real networks. Namely, this time we leverage the existence of *fat nodes* (see Definition 22) in preferential attachment graphs. We assume that they are globally known. The idea is that if we ask a known fat node for a friend, the received friend recommendation would have a more uniform distribution across all nodes. In consequence, the set of connections made by asking a fat node would create a subgraph which has a structure similar to a classic Erdös-Renyí graph.

k-Ask Fat For a Friend (k -A3F) goes as follows. Every participating node (similarly as with previous algorithm, due to various reasons it might be that some



(a) Enriched graph after attack (b) Initial graph after attack

Figure 3.4: Comparison of the outcome for adversarial attack example when 2-2SFF protocol was, and was not applied.

nodes do not want to participate actively) has to choose uniformly at random one fat node from the common list and ask for an address of one of its neighbors chosen at random. As previously, this is a procedure for each node v, initially $m = k$ if node is participating or $m = 0$ otherwise. Messages were explained in previous subsection. Formally, k -A3F protocol is presented as Algorithm 2.

See that in this protocol we rely on knowledge about existing fat nodes, ability to send a message to them and their willingness to perform the protocol for the benefit of the whole network.

3.2.3 k-Two Steps Fat Friend Finder

This protocol is in some sense a combination of both previous approaches. We want to use the existence of fat nodes but do not assume knowledge about them or ways to communicate with them. Instead of randomly choosing a friend to ask for recommendation, we want to introduce a bias towards asking friends with higher degree. That means nodes with more connections will be asked more frequently to mediate between two nodes. See that most of the nodes have connection to at least one fat node, which have substantially higher degree than other nodes. This will result in significant number of queries being sent to a fat node, which again results in more uniform structure of graph induced by new connections.

The participating node has to choose friend to get recommendation, as in 2SFF. However, this time it is not done uniformly, but based on the number of connections that its neighbor has. As before, this procedure is being independently

```
1  foreach node v do
2      while true do
3          if received msg = REQ <u> then
4              1. r ← rand(0, |N[v]|).
5              2. Send message "FRD <N[r]>" to node u.
6          else if received msg = FRD <u> then
7              1. Send message "EDG <v>" to node u.
8          else if received msg = EDG <u> then
9              1. Add u to list of neighbors.
10             2. Send message "ACK <v>" to node u.
11         else if received msg = ACK <u> then
12             1. Add u to list of neighbors.
13             2. m ← m - 1.
14         else if m > 0 then
15             1. r ← rand(0, |F|).
16             2. Send message "REQ <v>" to node F[r].
```

Algorithm 2: k -A3F

iterated k times. That would result in obtaining at most k new connections. Note however, that we give weights only to choose the mediating friend. Afterwards, this chosen friend gives the recommendation uniformly at random. Otherwise this algorithm would be prone to targeted attacks (see Section 3.4).

Algorithm k-Two Steps Fat Friend Finder (k-2S3F, for short) is presented as Algorithm 3. Again, this is a procedure for each node v, initially $m = k$ if node is participating or $m = 0$ otherwise. By W we denote the array of neighbors weights. Moreover, *randWeighted*(*V, W*) is a function that takes array of vertices V, array of their weights W and returns item randomly chosen from array of vertices with probability proportional to weights.

Note that here we assume only that each node knows their friends' number of connections and do not assume any global knowledge of the network.

3.3 Analytic Results

In this section we formally analyse some specific, most interesting cases of our protocols in a general model. Other cases are also considered in the next section, where we do experimental analysis of our protocols.

```
1  foreach node v do
2      while true do
3          if received msg = REQ <u> then
4              1. r ← rand(0, |N[v]|).
5              2. Send message "FRD <N[r]>" to node u.
6          else if received msg = FRD <u> then
7              1. Send message "EDG <v>" to node u.
8          else if received msg = EDG <u> then
9              1. Add u to list of neighbors.
10             2. Send message "ACK <v>" to node u.
11         else if received msg = ACK <u> then
12             1. Add u to list of neighbors.
13             2. m ← m - 1.
14         else if m > 0 then
15             1. u ← randWeighted(N[v], W).
16             2. Send message "REQ <v>" to node u.
```

Algorithm 3: k-2S3F

3.3.1 $\log n$-A3F under Targeted Attack

Let us analyse the $\log n$-A3F with Adversary knowing the topology of graph G in advance thus attacking the nodes with the highest degree. We consider $G = (V, E)$ to be generated as a preferential attachment graph. One of properties of such graphs is existence (whp) of a group of vertices (we call them *fat nodes*, see Section 3.1) having high degrees which combined neighborhoods cover whp the linear number of vertices from V.

Thus, let us assume throughout this subsection the following. Let $F \subset V$ be the subset of fat nodes, with fixed constants a and b (see definition in Section 3.1). Furthermore, as the Adversary uses Targeted Attack strategy, this is the set of vertices that will be corrupted by the Adversary. By N_F we denote the neighborhood of F without vertices from F, thus $N_F = \bigcup_{f \in F} N(f) \setminus F$. Let us also denote $\omega = \frac{|F|}{\log n}$. We assume also that $|V \setminus (F \cup N_F)| \leqslant \alpha n$ for some constant $0 < \alpha < 1$. Let $V_\alpha = V \setminus (F \cup N_F)$.

To begin our analysis, let us consider the case in which all vertices want to participate in the $\log n$-A3F Protocol.

Theorem 8. *If $\omega a < 1 - \alpha$ then after executing $\log n$-A3F for all vertices in*

$G = (V, E)$ *we obtain* $G_A = (V \setminus F, (E \cup E_P) \setminus E_C)$ *which is whp* 1*-strong. (Recall that* E_P *is the set of edges added during the protocol execution and* E_C *is the set of edges incident to vertices from* F*.)*

Proof Note that the set of vertices of G_A satisfies $V \setminus F = N_F \cup V_\alpha$. Moreover, N_F and V_α are disjoint. First, let us concentrate on the set N_F. Let $u, v \in N_F$. Let $f \in F$ be such that $\{f, v\} \in E$. Let us estimate the probability that there exists an edge $\{u, v\}$ (denote this event by $[u \leftrightarrow v]$). Let $[u \rightarrow v]$ denote the event that u established an edge $\{u, v\}$ during the protocol. For some $\varepsilon > 0$ and sufficiently big n we get

$$\begin{aligned} \mathbb{P}([u \rightarrow v]) \geq 1 - \left(1 - \frac{1}{\omega \log n} \frac{1}{deg(f)}\right)^{\log n} \geq \\ 1 - \left(1 - \frac{1}{\omega \log n} \frac{\log n}{an}\right)^{\log n} \geq \\ 1 - e^{-\frac{\log n}{\omega a n}} \geq \frac{\log n}{\omega a n + \log n} \geq (1 + \varepsilon) \log(|N_F|)/|N_F|. \end{aligned} \tag{3.1}$$

Note that $1/(\omega \log n \ deg(f))$ is the lower bound for the probability that v establishes an edge $\{v, u\}$ in a single step of the protocol. Indeed, f does not need to be the only neighbor of v in F. The second inequality follows from the bounds for $deg(f)$. The third and fourth inequalities follow from the fact that $(1 + x) \leqslant e^x$ for all $x \in \mathbb{R}$. The last inequality follows because $\omega a < 1 - \alpha$ and $|N_F| = (1 - \alpha)n - \omega \log n$. Since each vertex creates new edges during the protocol independently from other vertices, we have $\mathbb{P}([u \leftrightarrow v]) = \mathbb{P}([u \rightarrow v]) + \mathbb{P}([v \rightarrow u]) - \mathbb{P}([u \rightarrow v])\mathbb{P}([v \rightarrow u])$. Of course, the lower bound (3.1) is true also for $[v \leftrightarrow u]$ for all $u, v \in N_F$. We can think that the subgraph of G induced on N_F (denote it by $G(N_F)$) decomposes into Erdős-Renyí $G(N_F, p)$, where $p \geq (1 + \varepsilon) \log(|N_F|)/|N_F|$, and some remaining random graph. Thus $G(N_F)$ will inherit some monotone properties of $G(N_F, p)$, among others, it will be connected whp. Since the Adversary corrupts the nodes with the highest degrees, namely the whole set F, all the vertices from N_F will stay in G_A. Thus we have proved the existance (whp) of a giant component (which contains $G(N_F)$) of size at least $|N_F| = (1 - \alpha)n - \omega \log n$ in G_A.

Now, let us concentrate on the set V_α. Let us estimate the probability that a vertex $v \in V_\alpha$ is not connected with $G(N_F)$ (denote this event by $[v \not\leftrightarrow G(N_F)]$). What needs to happen is that whenever the fat node sends to v the id of u, u needs

to be a fat node as well. Since there are $\omega \log n$ fat nodes and their degrees are at least $an/\log n$, we obtain

$$\mathbb{P}([v \not\leftrightarrow G(N_F)]) \leq \left(\frac{\omega \log n}{an/\log n}\right)^{\log n} = \left(\frac{\omega(\log n)^2}{an}\right)^{\log n}.$$

Vertices from V_α act during the protocol independently and the above probability is vanishing, so we can simply estimate the probability that all vertices from V_α are connected with $G(N_F)$ (denote this event by $[V_\alpha \leftrightarrow G(N_F)]$) and show that it happens whp:

$$\mathbb{P}([V_\alpha \leftrightarrow G(N_F)]) \geq \left(1 - \left(\frac{\omega(\log n)^2}{an}\right)^{\log n}\right)^{\alpha n} \xrightarrow{n\to\infty} 1.$$

Thus whp G_A is connected. □

Now, let us discuss the following case: β fraction of vertices from V_α and γ fraction of vertices from N_F take part in the protocol. Note that we do not care about vertices from F because they are going to be corrupted and their incident edges will not appear in G_A eventually.

Theorem 9. *If $\omega a < 1 - \alpha$ and $\omega b > \gamma(1-\alpha)$ then after executing $\log n$-A3F for vertices as described above on $G = (V, E)$ we obtain $G_A = (V \setminus F, (E \cup E_P) \setminus E_C)$ which is whp $(1 - (1-\beta)\alpha)$-strong.*

Proof Let $\tilde{N}_F$ denote the set of vertices from N_F which take part in the protocol ($|\tilde{N}_F| = \gamma|N_F|$). Even though the vertices from $N_F \setminus \tilde{N}_F$ do not take part in the protocol, they can be chosen as those to whom vertices from $\tilde{N}_F$ establish new edges. Let us estimate the probability that $v \in N_F \setminus \tilde{N}_F$ will not get connected to any vertex from $\tilde{N}_F$ during the execution of the protocol (denote this event by $[v \not\leftrightarrow G(\tilde{N}_F)]$). Let f be such that f and v are neighbors in G. We have

$$\mathbb{P}([v \not\leftrightarrow G(\tilde{N}_F)]) \leq \left(1 - \frac{1}{\omega \log n \cdot deg(f)}\right)^{\gamma|N_F|\log n} \leq \left(1 - \frac{1}{\omega bn}\right)^{\gamma|N_F|\log n} \leq$$
$$\leq e^{-(\gamma \log n|N_F|)/(\omega bn)} = n^{-\gamma(1-\alpha)/(\omega b)} n^{\log n/(bn)}$$

(compare 3.1).

Now, let us estimate the probability that all vertices from $(N_F \setminus \tilde{N}_F)$ are going to be connected with $G(\tilde{N}_F)$ (denote this event by $[(N_F \setminus \tilde{N}_F) \leftrightarrow G(\tilde{N}_F)]$). We get

$$\mathbb{P}([(N_F \setminus \tilde{N}_F) \leftrightarrow G(\tilde{N}_F)]) \geq \left(1 - n^{-\gamma(1-\alpha)/(\omega b)} n^{\log n/(bn)}\right)^{(1-\gamma)|N_F|} =$$
$$= \left(1 - n^{-\gamma(1-\alpha)/(\omega b)} n^{\log n/(bn)}\right)^{(1-\gamma)((1-\alpha)n - \omega \log n)} \xrightarrow{n\to\infty} 1$$

since $\omega b > \gamma(1-\alpha)$. Thus again whp $G(N_F)$ is connected.

By calculations analogous to those from Theorem 8 we also get that all vertices from V_α which participate in the protocol (denote this set by $\tilde{V}_\alpha$) are connected with $G(N_F)$ whp. We proved that whp G_A has a giant component containing $N_F \cup \tilde{V}_\alpha$, such that $|N_F \cup \tilde{V}_\alpha| = (1-\alpha)n - \omega \log n + \beta\alpha n$. This completes the proof.

□

3.3.2 $\log n - 2S3F$ under Targeted Attack

Below we present the formal analysis of $\log n - 2S3F$ protocol. Throughout this subsection we assume that every node applies the security protocol and that we handle again with the Adversary knowing the topology of the network thus attacking fat nodes. We again formally represent our network by preferential attachment graph, however now we set $m = 2$ (see Definition 2).

The results can only be better if we dealt with denser graphs (for $m > 2$). We omit the case $m = 1$, since then the resulting structure of preferential attachment process is simply a tree.

Recall that we denote the set of fat nodes by F. In this subsection we assume that $|F| = \omega \log n$ for some constant ω. It is a justified assumption about preferential attachment structure which follows from its scale-free property. The set of fat nodes F is, as in previous subsection, being corrupted by the Adversary. We also assume again that the neighborhood of fat nodes (we denote it by $N(F)$) covers whp the linear number of vertices from V.

Throughout this subsection we are going to use Fact 1 and Theorem 1 from Subsection 1.4.2. What we want to prove in this subsection is that the structure of preferential attachment graph with parameter $m = 2$ is whp ξ-strong for some constant $\xi \in (0,1)$ after applying $\log n - 2S3F$ protocol if we struggle with the Adversary corrupting fat nodes. First, let us state a few auxiliary lemmas.

Lemma 3. *Let F denote the set of fat nodes and let $f \in F$. Let $C_f \subset N(f)$ be a set such that $v \in C_f$ if and only if $deg(v) \leq (\log n)^{1-\varepsilon}$ for some $\varepsilon > 0$. Then*

$$\mathbb{E}|C_f| \sim \left(1 - \frac{2}{(\log n)^{1-\varepsilon}}\right) \frac{dn}{\log n}$$

for some $d \in [a, b]$.

Remark 1. Of course, if $\mathbb{E}|C_f| \sim \left(1 - \frac{2}{(\log n)^{1-\varepsilon}}\right) \frac{dn}{\log n}$ we can write simply $\mathbb{E}|C_f| \sim \frac{dn}{\log n}$. However, later on we will be interested also in the expected size of the leftover, namely $\mathbb{E}[|N(f) \setminus C_f|] \sim \frac{2}{(\log n)^{1-\varepsilon}} \frac{dn}{\log n}$. Therefore we leave the factor $\left(1 - \frac{2}{(\log n)^{1-\varepsilon}}\right)$ in Lemma 3.

Proof Since f is a fat node we have $deg(f) = |N(f)| = dn/\log n$ for some $d \in [a, b]$. We work with a preferential attachment graph with parameter $m = 2$, so every vertex has degree at least 2. We have

$$\mathbb{E}|C_f| = |N(f)| \sum_{l=2}^{\lfloor (\log n)^{1-\varepsilon} \rfloor} p(l|deg(f)),$$

where $p(l|deg(f))$ is the probability that a randomly chosen neighbor of f will have degree l. By Theorem 1 and the fact that $\binom{deg(f)+l-4}{l-2} \sim \frac{(deg(f)+l-4)^{l-2}}{(l-2)!}$ and $\binom{deg(f)+l+2}{l} \sim \frac{(deg(f)+l+2)^l}{l!}$ we get

$$\begin{aligned} p(l|deg(f)) &\sim \frac{2(deg(f)+2)}{deg(f)l(l+1)} \left(1 - \frac{l+1}{deg(f)+2} \binom{6}{3} \frac{\binom{deg(f)+l-4}{l-2}}{\binom{deg(f)+l+2}{l}}\right) \\ &\sim \frac{2}{l(l+1)} \left(1 - \frac{20(l-1)l(l+1)}{deg(f)^3}\right) \sim \frac{2}{l(l+1)}, \end{aligned}$$

thus

$$\mathbb{E}|C_f| \sim \frac{dn}{\log n} \sum_{l=2}^{\lfloor (\log n)^{1-\varepsilon} \rfloor} \frac{2}{l(l+1)}.$$

Since $\sum_{l=2}^{\infty} \frac{2}{l(l+1)} = 1$ and $\sum_{l=\lceil (\log n)^{1-\varepsilon} \rceil}^{\infty} \frac{2}{l(l+1)} = 2/\lceil (\log n)^{1-\varepsilon} \rceil$ we get

$$\mathbb{E}|C_f| \sim \left(1 - \frac{2}{(\log n)^{1-\varepsilon}}\right) \frac{dn}{\log n}.$$

□

Remark 2. Note that we always have $|C_f| \leq \frac{dn}{\log n}$. On the other hand the neighborhood of f is a significant part of the whole graph, thus partly preserving the characteristics of scale-free structures. Among others, the fraction of vertices of degree k goes for large values of k as $k^{-\lambda}$ for some constant $\lambda > 1$, which means that the number of nodes of degree greater than $(\log n)^{1-\varepsilon}$ (for some $\varepsilon > 0$) in $N(f)$ is negligible. Therefore we can write not only $\mathbb{E}|C_f| \sim \frac{dn}{\log n}$ but even $|C_f| \sim \frac{dn}{\log n}$ whp.

Lemma 4. *Let F denote the set of fat nodes and let $f \in F$. Let C_f be defined as in Lemma 3. Let G_{C_f} be the graph induced on the set C_f after applying the $\log n - 2S3F$ protocol . Then G_{C_f} is whp connected.*

Proof Let $u, v \in C_f$. Let us estimate $\mathbb{P}[u \to v]$ - the probability that u has established a new connection with v while applying the $\log n - 2S3F$ protocol. Note that the probability that it happens in a single query is at least $1/((\log n)^{1-\varepsilon} deg(f))$ (indeed, u is of degree at most $(\log n)^{1-\varepsilon}$, so $1/(\log n)^{1-\varepsilon}$ is the lower bound for the probability that u chooses f while $1/deg(f)$ is the probability that f sends back v as his friend). Since $deg(f) = dn/\log n$ for some $d \in [a, b]$ and u sends $\log n$ independent queries, we get

$$\mathbb{P}[u \to v] \geq 1 - \left(1 - \frac{1}{(\log n)^{1-\varepsilon} deg(f)}\right)^{\log n} = 1 - \left(1 - \frac{(\log n)^{\varepsilon}}{dn}\right)^{\log n}$$
$$\geq 1 - (1/e)^{(\log n)^{1+\varepsilon}/dn} \geq 1 - \frac{1}{1 + (\log n)^{1+\varepsilon}/dn} = \frac{(\log n)^{1+\varepsilon}}{dn + (\log n)^{1+\varepsilon}}.$$

Note that the connections that appear during the protocol execution are established independently from each other. Therefore we can think of them as of the edges of Erdös-Renyí graph $G(|C_f|, p_n)$ with $p_n \geq \frac{(\log n)^{1+\varepsilon}}{dn+(\log n)^{1+\varepsilon}}$. For sufficiently large n we have $p_n \geq \frac{(\log n)^{1+\varepsilon}}{dn+(\log n)^{1+\varepsilon}} \geq \frac{(1+\varepsilon)\log n}{n}$. Thus by Fact 1 $G(|C_f|, p_n)$ is whp connected (indeed, by Remark 2 we have whp $|C_f| \sim \frac{dn}{\log n}$). □

What we get from Lemma 3 and Lemma 4 is that if we look at the neighborhood of any fat node f (which is of the size $dn/\log n$ for some $d \in [a, b]$) after the $\log n - 2S3F$ protocol execution, we will find there whp a subgraph C_f of the size $\sim \frac{dn}{\log n}$ which is connected. We will call the set C_f a *cloud* of f. What we are going to show next is that the clouds of all fat nodes intersect and that their total size is whp linear in n. This will give us the desired ξ-strength - even if the Adversary disables all fat nodes, we are left with intersecting clouds forming connected subgraph of size which is whp linear in the number of honest nodes.

Lemma 5. *The number of vertices belonging to all the clouds of fat nodes is whp linear in n.*

Proof By the assumptions of this subsection, the neighborhood of all fat vertices covers whp the linear number of vertices in V. We have whp $|N(F)| = \gamma n$ for some positive constant $\gamma \in (0,1)$. For any fat node f of degree $dn/\log n$ ($d \in [a,b]$) we have by Lemma 3 and Remark 2 that the number of vertices not belonging to the cloud of f is whp $\sim \frac{2}{(\log n)^{1-\varepsilon}} \frac{dn}{\log n}$. The number of clouds is $\omega \log n$ for some constant ω and $bn/\log n$ is the upper bound for the degree of any fat node. We get that the number of vertices belonging to all the clouds of fat nodes is whp at least $\sim \gamma n - \omega \log n \frac{2}{(\log n)^{1-\varepsilon}} \frac{bn}{\log n}$ which is linear in n. □

Theorem 10. *The clouds of all fat nodes from F form whp a connected subgraph.*

Proof We are going to show that the statement of this theorem follows from the process of building the preferential attachment graph. The whole structure is built in n steps (where one step is adding one vertex to the graph). After $n/2$ steps we can already see the structure whose characteristics are analogous to those in the full structure, among others, we already have $\sim \log n$ fat nodes, each of degree at most $bn/\log n$. Now, let us consider those vertices from the cloud C_f which arrived after time $n/2$. Recall that by Remark 2 the number of vertices in C_f is whp $\sim \frac{dn}{\log n}$ (f is of degree $dn/\log n$). We assume that at least $\delta \frac{dn}{\log n}$ of them appeared after step $n/2$, where δ is some positive constant. Note that each of those vertices has at least one more edge apart from the one connecting it to f. With probability at least $\frac{\beta}{\log n}$ for some constant β it is the edge attaching it to the other particular fat node (after step $n/2$ every fat node has already the degree of the order $\frac{n}{\log n}$ and the sum of all degrees is already linear in n).

Now, we will show that vertices of C_f which came after step $n/2$ are whp attached to all the other clouds as well. This will finish the proof. We have $\omega \log n$ clouds, name them $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$. At least $\delta \frac{dn}{\log n}$ vertices from C_f come after step $n/2$ and the probability that such a vertex attaches also to the other particular cloud is at least $\beta/\log n$. For $j \in \{1,2,\ldots,k\}$, the probability that C_f will not be connected with C_{v_j} is at most

$$\left(1 - \frac{\beta}{\log n}\right)^{\delta \frac{dn}{\log n}} < (1/e)^{\frac{\beta \delta dn}{(\log n)^2}}.$$

By the union bound the probability that there exists $j \in \{1, 2, \ldots, k\}$ such that C_f is not attached to C_{v_j} is at most

$$\omega \log n (1/e)^{\frac{\beta\delta dn}{(\log n)^2}} \xrightarrow{n\to\infty} 0.$$

Thus the probability that C_f is attached to all the clouds is at least

$$1 - \omega \log n (1/e)^{\frac{\beta\delta dn}{(\log n)^2}} \xrightarrow{n\to\infty} 1.$$

□

Corollary 4. *The structure of preferential attachment graph with $m = 2$ after applying $\log n - 2S3F$ protocol is whp 1-strong by the attack of targeted Adversary who corrupts all fat nodes.*

3.4 Experimental Results

We present experimental results conducted on **real** data of Epinions social network collected in SNAP dataset by Stanford University (see [44] and [58]).

This is a who-trust-whom online social network of a general consumer review site Epinions.com. Members of the site can decide whether to "trust" each other. All the trust relationships interact and form the Web of Trust which is then combined with review ratings to determine which reviews are shown to the user. This network has $75,879$ nodes and $508,837$ edges where nodes denote users of Epinions.com site and edges denote trust relation.

3.4.1 Random Failures

First we consider the Random Failures model. We assume that corrupted nodes (or in other words, nodes which are prone to failure) are distributed in a uniform way across the whole network.

k-A3F Protocol

Initially we assume that all nodes participate in the protocol, namely each node does k queries which consist of randomly choosing one of the fat nodes and asking for randomly chosen neighbor of that node. Obviously, the larger k, the higher

safety of the nodes. Here we fixed the number of the nodes considered fat to $\lfloor \log(n) \rfloor = 16$. It means that 16 nodes which have the highest degree in the initial graph are on the common list of 'fat nodes'. Note that due to real network properties (namely preferential attachment) these nodes have significantly higher degree than the average node degree in this graph.

In Figure 3.5 one can see the performance of A3F on Epinions social network graph under Random Failures model. We can see how the network behaves without any enrichment, and with $k = 1, 5, 10, 15$. Note that on the x-axis we have the percentage of corrupted nodes. With $k = 15$ queries, almost 90% of remaining nodes are in the largest connected component despite a large number of failures.

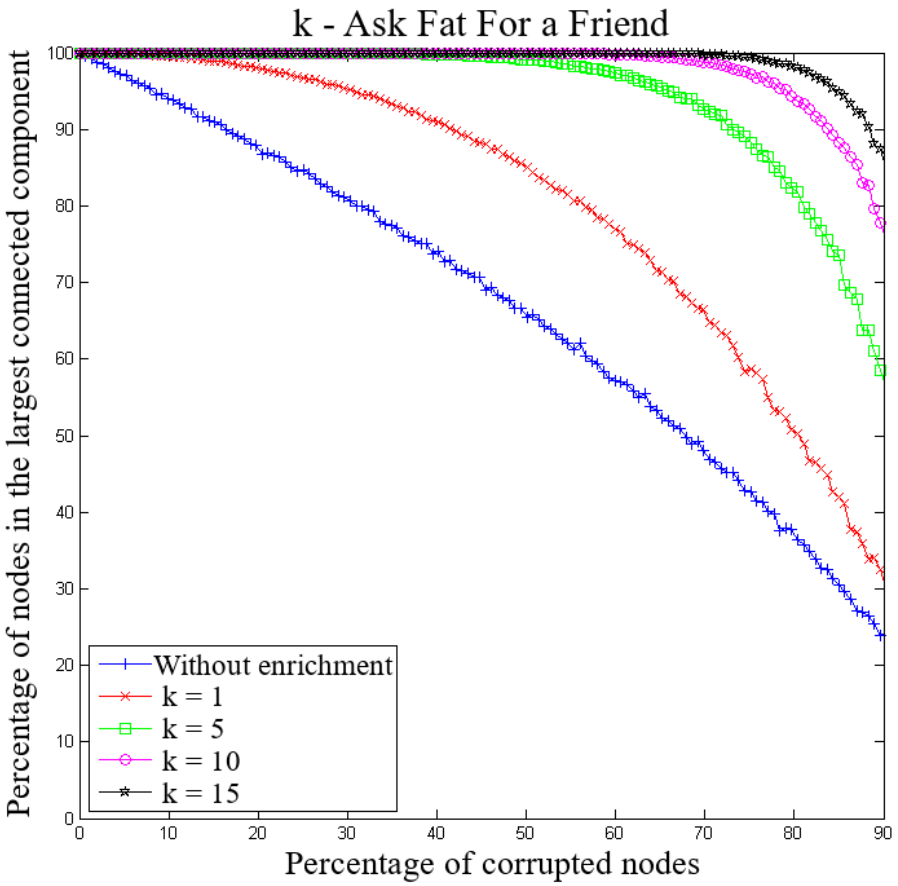


Figure 3.5: k-A3F under Random Failures model.

Despite these somewhat optimistic results, it is quite unrealistic to assume that all users want to participate, so we want to weaken this assumption. We still demand high level of security, at least for the participating users. We assumed $k = 15$ and $q = 0.1, 0.25, 0.5$ fractions on nodes participating. In Figure 3.6 we have shown the results for 15-A3F with partial participation.

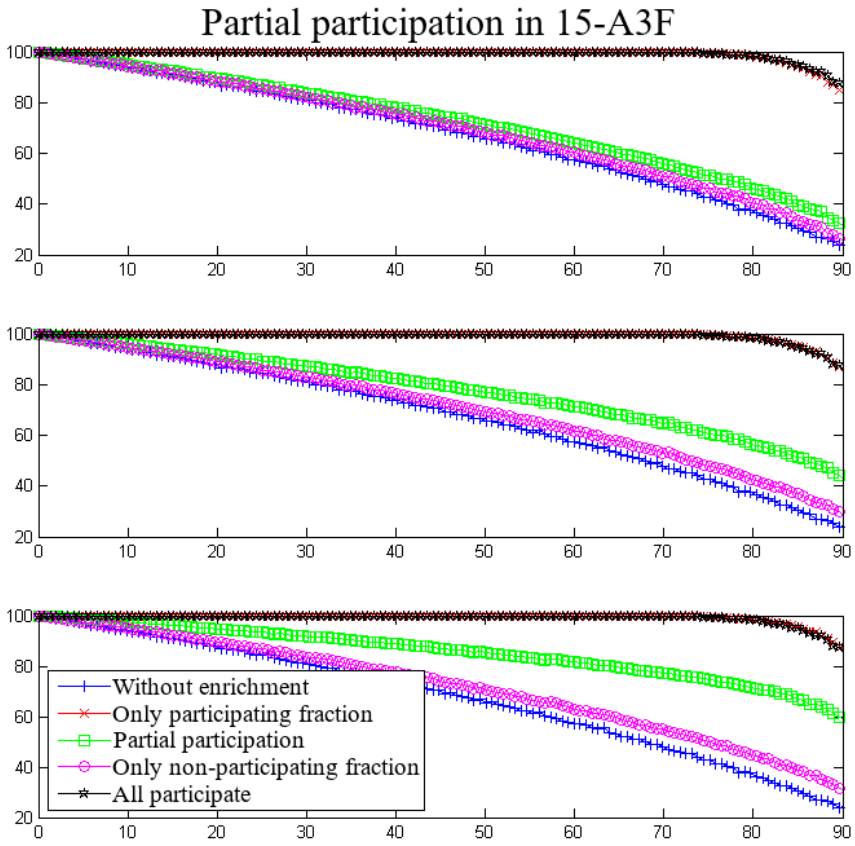


Figure 3.6: 15-A3F under Partial Participation and Random Failures model. The top figure shows 10% participation, middle shows 25% participation and bottom 50% participation.

Security implications The most interesting thing is the fact that the safety level amongst the participating nodes in case of partial participation is virtually the

same as the safety level when all nodes participate. This fact is very important from the practical point of view. It gives users a choice - whether they want to sacrifice their safety and not participate in the protocol, or participate in the protocol and be safe no matter what other users choose as long as at least some fraction (say 10%) decides to participate in the protocol.

k-2SFF Protocol

Now we focus on the k-2SFF Protocol in the case of Random Failures. Initially we assume that all nodes launch the k-2SFF Protocol, namely each node does k random walks of length 2 to establish extra connections.

In Figure 3.7 we show how the k-2SFF Protocol performs on Epinions social network graph under Random Failures model. Similarly as before, we show the behavior of the network without any enrichment, and with $k = 1, 5, 10, 15$. Note that on the x-axis we have the percentage of corrupted nodes. This time, with $k = 15$, around 70% of remaining nodes are in the largest connected component even if up to 90% vertices were corrupted.

Now we are interested in the performance of 2SFF in the case where only a fraction of users wants to participate. In Figure 3.8 we show some experimental results when a part of nodes participates, only. Here we assume $k = 15$ and $q = 0.1, 0.25, 0.5$ fraction of participating nodes. That is, $q \cdot n$ nodes participate in 15-2SFF protocol. Then we are interested what is the fraction of participating users that belong to the biggest component and how it compares to the situation when all users do participate.

Note that in the case where $q = 0.1$ there is a significant decrease of security. Namely, with massive number of failures, we have around 30% nodes in biggest component in comparison to 70% in the full participation case. Note that even if we consider only the subset of participating nodes, then the fraction of nodes belonging to biggest component amongst them is below 40%. The security indeed improves with greater q, yet still even if we consider only the participating nodes, the results are significantly worse than when all users participate. Thus this protocol turned out to be useful in communities if we know that strong majority of nodes is willing to use it.

k-2S3F Protocol

This subsection is devoted to k-2S3F Protocol for Random Failures case. As in previous experiments, first we assume that all nodes actively participate in the pro-

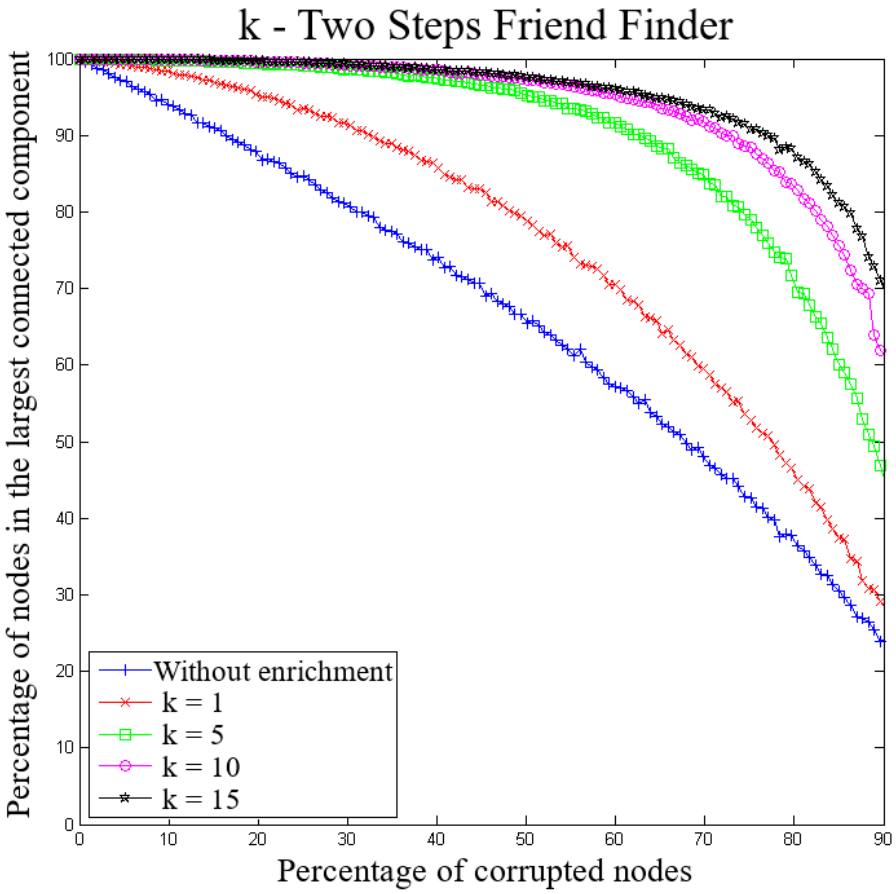


Figure 3.7: k-2SFF under Random Failures model.

tocol, namely each node chooses a neighbor (neighbor with more connections has higher probability due to weights) and chosen neighbor sends him a connection, this time chosen uniformly at random.

In Figure 3.9 we show how the k-2S3F Protocol performs on Epinions social network graph under Random Failures model. We show the network without any enrichment, and with $k = 1, 5, 10, 15$. Note that on the x-axis we have the percentage of corrupted nodes. With $k = 15$, almost 80% of remaining nodes are in the largest connected component even with 90% corrupted nodes. One can easily see that for up to around 20% failures, even 5 iterations of protocol are sufficient to have almost every node belonging to the largest connected component.

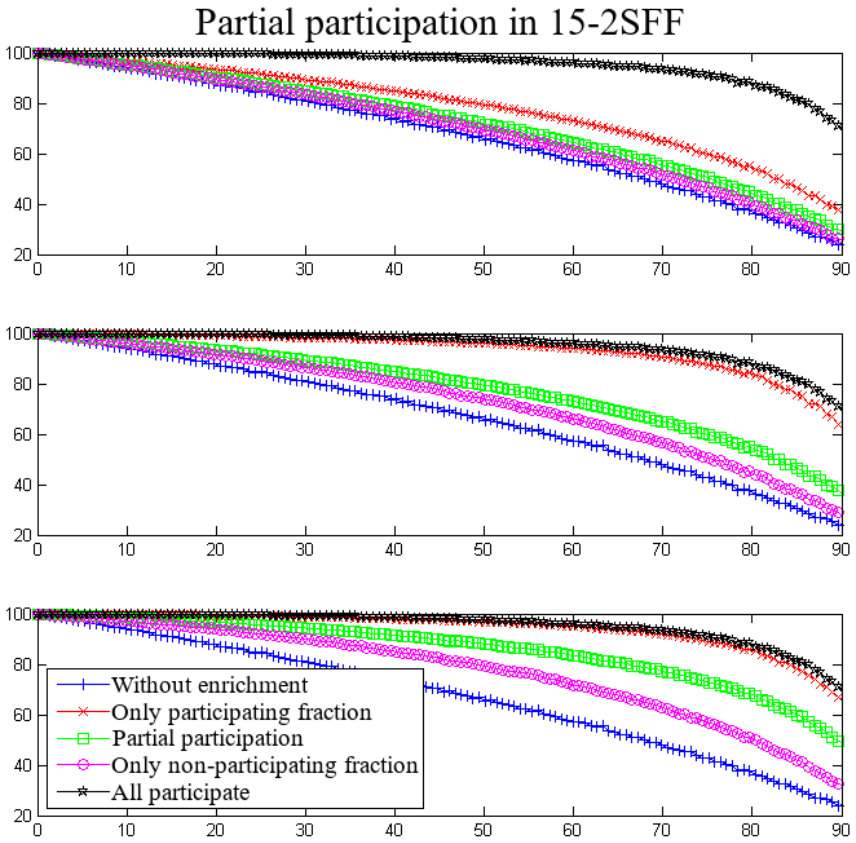


Figure 3.8: 15-2SFF under Partial Participation and Random Failures model. The top figure shows 10% participation, middle shows 25% participation and bottom 50% participation.

Comparison

After presenting all protocols under Random Failures regime, here we show a comparison of all these approaches. We pick $k = 15$, and compare A3F, 2SFF and 2S3F. We also consider a combined approach of 5 - A3F and 10 - 2S3F. This decreases pressure on fat nodes, which might be desired in systems where fat nodes do not have appropriate resources. The number of asked connections is 15, so this approach has exactly the same budget as others, yet it tries to leverage benefits of both A3F and 2S3F. It turned out to be a very promising strategy.

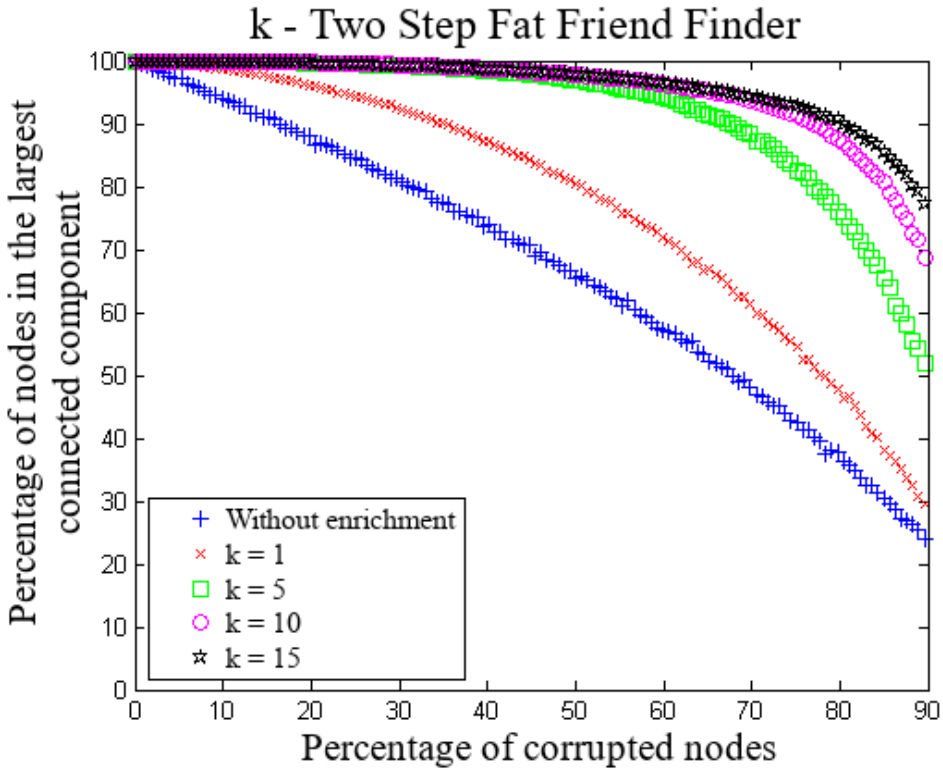


Figure 3.9: k-2S3F under Random Failures model.

In Figure 3.10 we show how all protocols behave under Random Failures. See that the best one is obviously A3F, yet combination of A3F and 2S3F turns out to be almost as effective, yet with less dependency on fat nodes.

A glance at the figures in this subsection is enough to see that k-A3F performs better than other protocols under Random Failures regime. See for example that for 90% failures the A3F protocol gives approximately 85% nodes belonging to the largest connected component, while 2SFF gives only 75%. Moreover, the cutoff and therefore non-negligible deterioration of the fraction of nodes in the biggest component happens for greater fraction of failures than in k-2SFF or k-2S3F protocol.

Intuitively, these differences in the results stem from the fact that in A3F we leverage naturally emerging preferential attachment models in real, complex networks, while 2SFF does not really utilize this fact. Connecting to neighbors of fixed, high-degree set of nodes massively improves robustness of real networks.

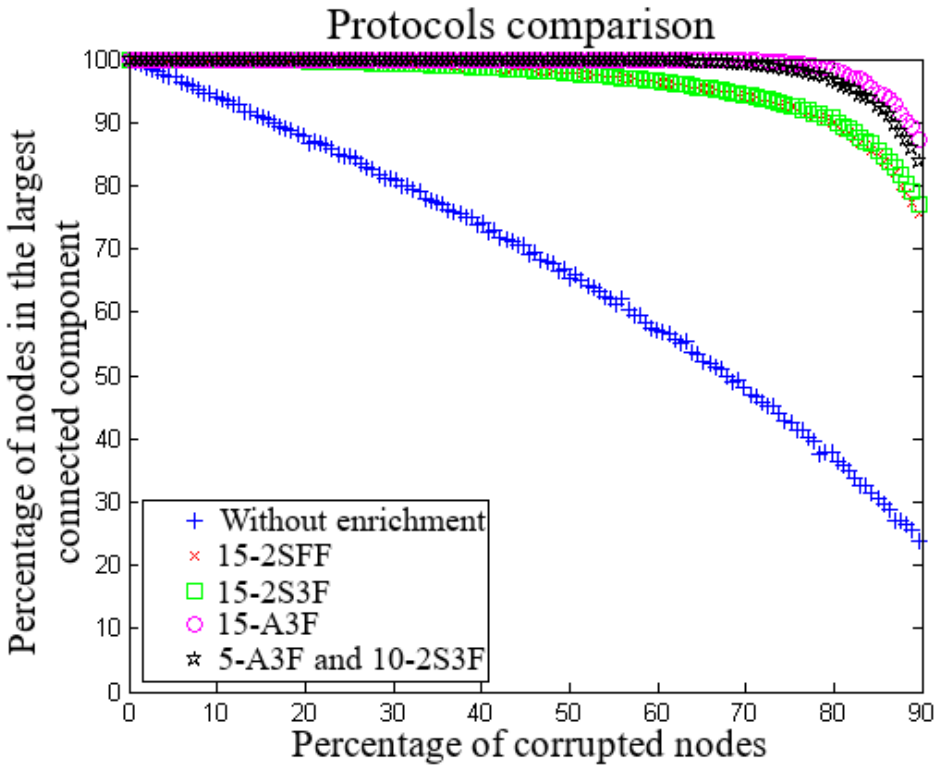


Figure 3.10: Comparison of all protocols for $k = 15$.

3.4.2 Targeted Adversary

In this subsection we present experiments conducted under far stronger adversary that can corrupt nodes of the highest degree. Namely, the adversary sorts the list of nodes by degree and corrupts k of them with largest degrees.

Note that the adversary only has access to the initial graph, before enrichment. Obviously, for a specific instance of the graph one could possibly devise a more clever way of attack, however this strategy seems to be optimal in general. Note that complex networks which resemble preferential attachment features are extremely prone to such attacks.

k-A3F protocol

In Figure 3.11 one can see the performance of k-A3F on Epinions social network graph under Targeted Adversary model. As previously, we show the behavior of

the network without any enrichment, and for the cases where $k = 1, 5, 10, 15$. Note that on the x-axis we have the percentage of corrupted nodes and this time it ranges from 0 to 30% instead of $0 - 90\%$ due to the Adversary's strength. This time, with $k = 15$ queries, approximately 85% of remaining nodes are in the biggest component for up to 30% corruptions and over 95% of nodes are in the largest connected component for up to 15% corruptions.

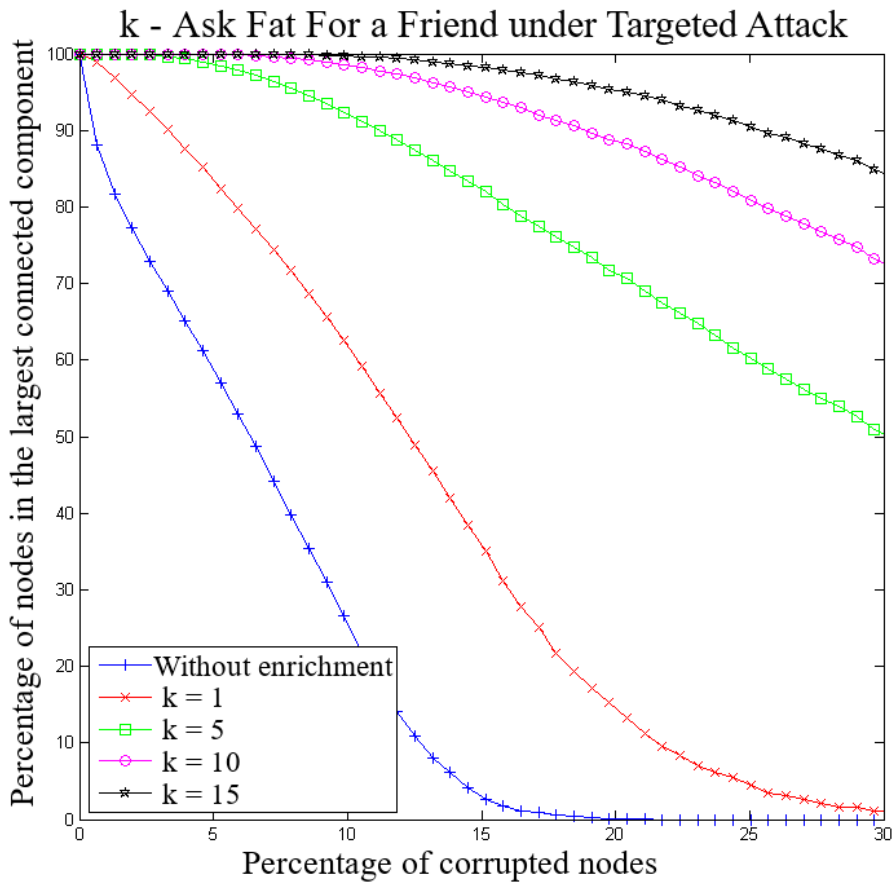


Figure 3.11: k-A3F under Targeted Attack.

As previously, we want to see how the protocol behaves if we assume that only a fraction of non-corrupted users participate actively. We assumed $k = 15$ and $q = 0.1, 0.25, 0.5$ fraction of nodes participating. In Figure 3.12 we show the results for A3F with partial participation under Targeted Adversary regime.

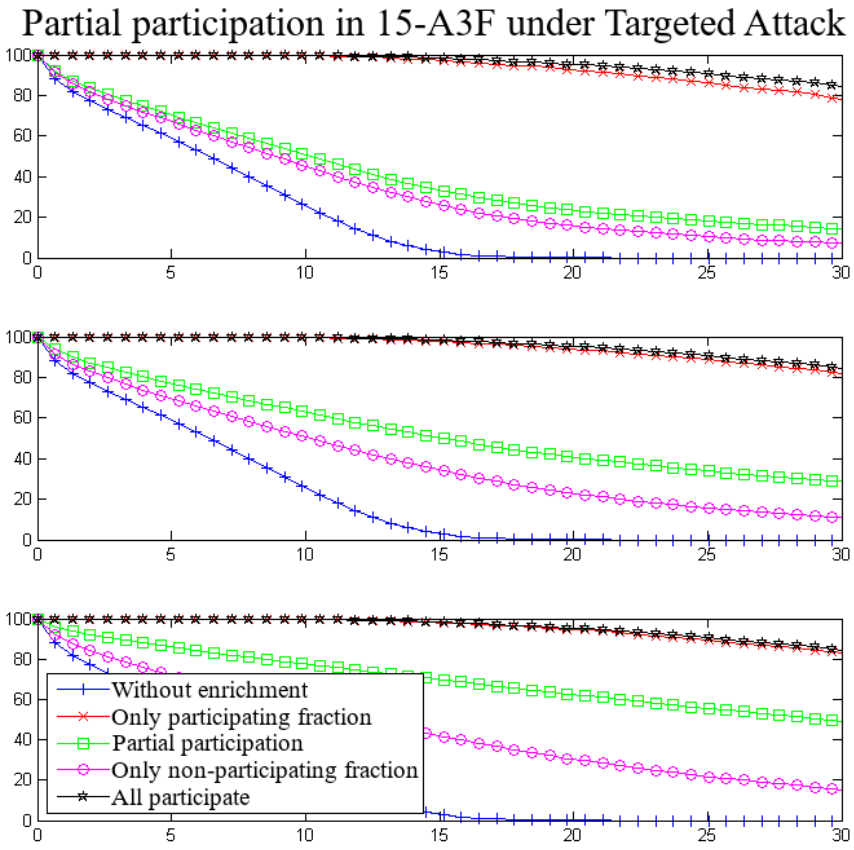


Figure 3.12: Partial 15-A3F under Targeted Attack. The top figure shows 10% participation, middle shows 25% participation and bottom 50% participation.

Figure 3.12 is probably the most striking one due to the fact that in all three cases, one can easily see that the fraction of nodes belonging to the largest connected component amongst the actively participating nodes is almost the same as when all nodes participate. This is a **very desirable** feature of k-A3F because it gives the user a natural choice - participate in the protocol, which costs some computational resources, but be in the largest connected component independently of the choices of other nodes or do not participate, but then you are facing serious risk of ending up disconnected from the largest connected component.

k-2SFF protocol

In Figure 3.13 we show how k-2SFF performs on Epinions social network graph under Targeted Adversary model. We can see how the network behaves without any enrichment, and with $k = 1, 5, 10, 15$. Note that without enrichment the fraction of nodes in the largest connected component falls to almost 0 for 20% failures. In other words, if the adversary destroys 20% nodes of highest degree, the remaining graph consists only of very small components. On the other hand, see that for up to 5% corruptions, $k = 15$ gives almost 100% nodes in the biggest component. Even for 30% corruption, the fraction of nodes in the biggest component is considerably large (approximately 60%). Recall that without enrichment under such a strong adversary there is virtually no large connected component whatsoever.

Let us investigate the protocol if we assume that only a fraction of non-corrupted users participate actively. We assumed $k = 15$ and $q = 0.1, 0.25, 0.5$ fraction of nodes participating. In Figure 3.14 we have shown the results for k-2SFF with partial participation under Targeted Adversary regime.

An interesting difference between the results for this model and Random Failures can be seen in this figure. Namely, the fraction of nodes belonging to the largest connected component amongst those who participate is only slightly greater than amongst those who do not participate. This is highly undesired, as it gives no notion of improvement and benefit of participating actively in the protocol. A node could decide that it is pointless to waste precious resources and rather hope that the others would participate actively. See that even if half of the users actively participate, the fraction of nodes in the largest connected component are significantly smaller than when all nodes participate.

These results for k-2SFF under Targeted Adversary are somewhat unsatisfactory, yet, as mentioned previously, this kind of adversary is extremely powerful for networks based on preferential attachment, and performing k-2SFF does not change graph structure strongly enough to defend against this kind of attack.

k-2S3F Protocol

In Figure 3.15 we show how the k-2S3F Protocol performs on Epinions social network graph under Targeted Attack adversarial model. We present the network without any enrichment, and with $k = 1, 5, 10, 15$. With $k = 15$, around 65% of remaining nodes are in the largest connected component for 30% corrupted nodes. On the positive side, for up to 10% failures, most of the remaining nodes belong to the largest connected component even under such a strong adversary.

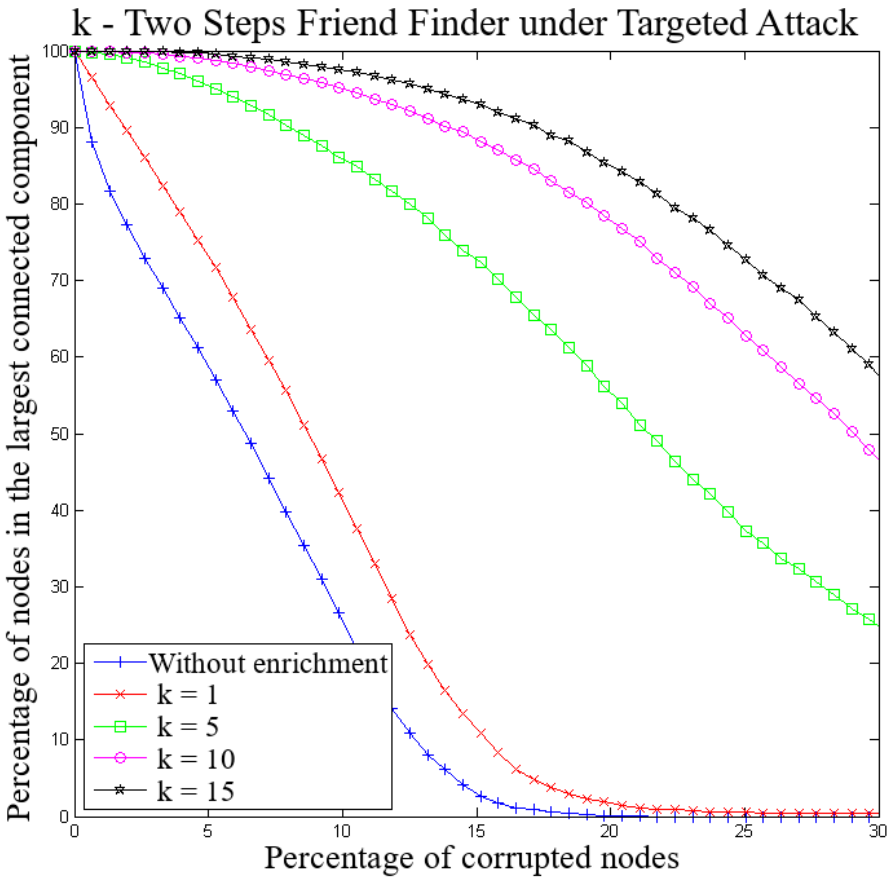


Figure 3.13: k-2SFF under Targeted Attack.

Now we assume that only a fraction of non-corrupted users participate actively. We assumed $k = 15$ and $q = 0.1, 0.25, 0.5$. In Figure 3.16 we have shown the results for k-2S3F with partial participation under Targeted Adversary regime.

We can see that similarly as in 2SFF, results are not very satisfying. On the positive side, for 50% participation we can see similar behaviour as in A3F, namely percentage of safe nodes amongst those who participate is similar as in the situation when all participate.

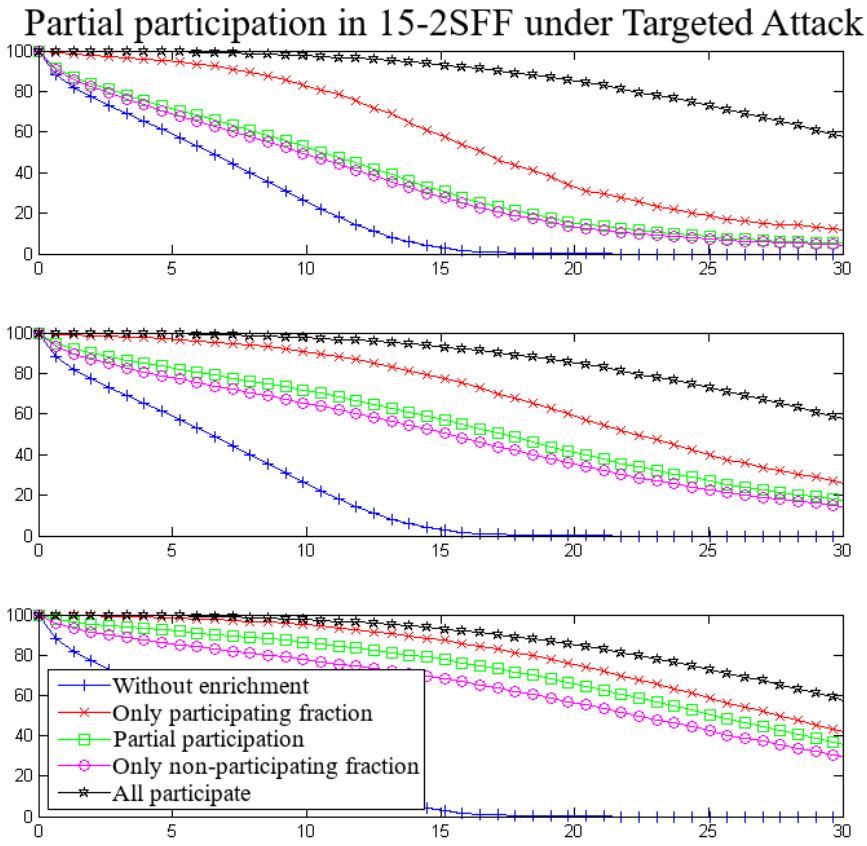


Figure 3.14: Partial 15-2SFF under Targeted Attack. The top figure shows 10% participation, middle shows 25% participation and bottom 50% participation.

Comparison

After presenting all protocols under Targeted Attack regime, we show a comparison of all these approaches. We assume $k = 15$, and compare A3F, 2SFF and 2S3F. Moreover, we also consider a combined approach of 5 - A3F and 10 - 2S3F. Note that this decreases pressure on the fat nodes, which might be desired, especially in systems where fat nodes do not have appropriate resources. One can see that, the number of asked connections is 15, so this approach has exactly the same budget as others, yet tries to leverage the advantages of both A3F and 2S3F strategies.

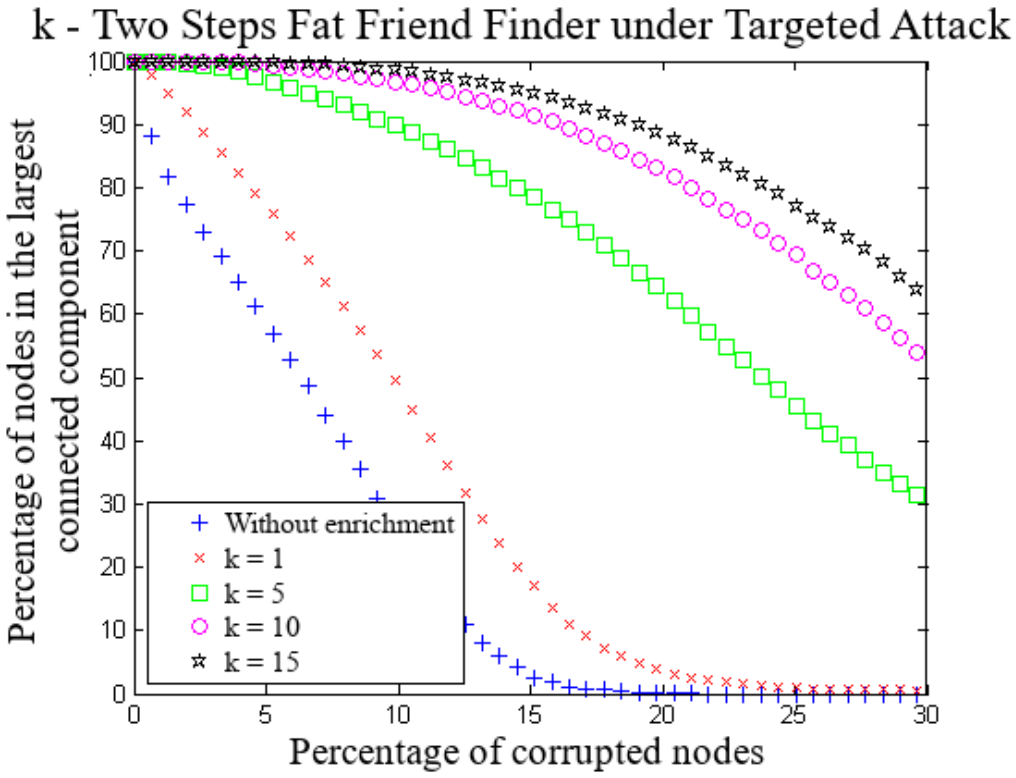


Figure 3.15: k-2S3F under Random Failures model.

In Figure 3.17 we show how all protocols behave under Targeted Attack. See that this time, the combination of A3F and 2S3F turned out to be the most effective strategy.

Of course the results for both protocols are obviously worse than for Random Failure model, which is not surprising. However, they still give a significant improvement of the size of the largest connected component. Moreover, in the regime of Targeted Adversary, the k-A3F has a very interesting property of assuring almost the same fraction of nodes belonging to the largest connected component for participating fraction of nodes (even if only 10% of users participate) as in the case where all users participate.

This regime shows that k-A3F is indeed a very powerful enrichment to the graph structure. Moreover, combining it with 2S3F makes it even more powerful. Note that we went from no large connected component for 20% failures to almost 90% nodes belonging to the largest connected component amongst the actively

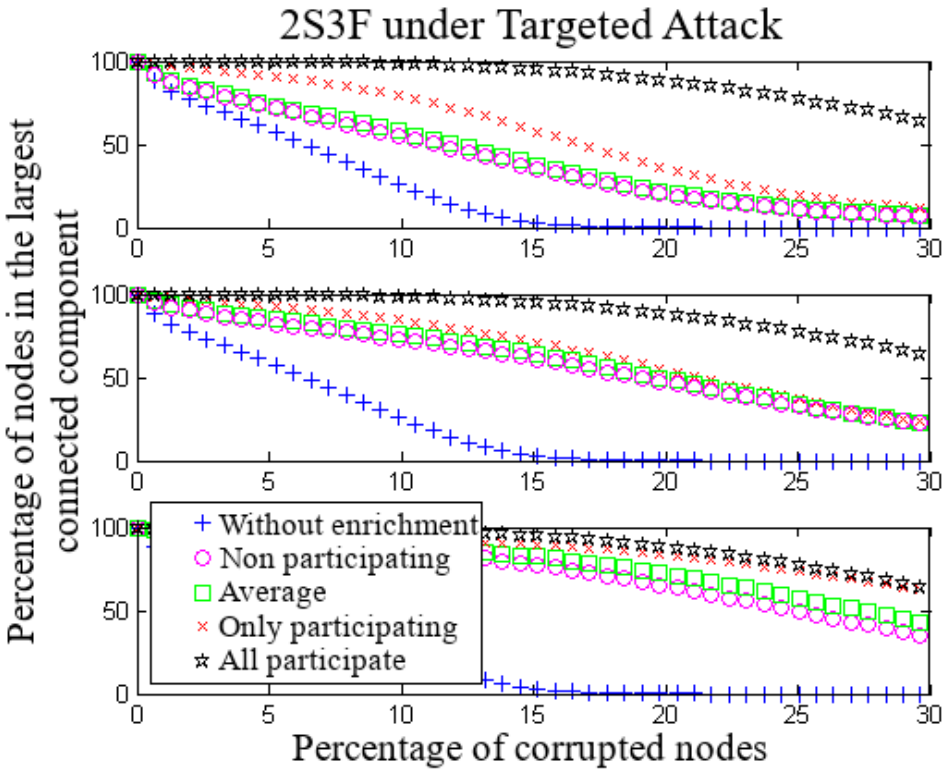


Figure 3.16: Partial 15-2S3F under Targeted Attack. The top figure shows 10% participation, middle shows 25% participation and bottom 50% participation.

participating nodes even if only 10% of users participate. This scenario shows a significant improvement of security which is gained via our protocol for those who actively participate in it. Note that the difference between the performance of k-2SFF and k-A3F is strongly connected with utilizing preferential attachment in real networks.

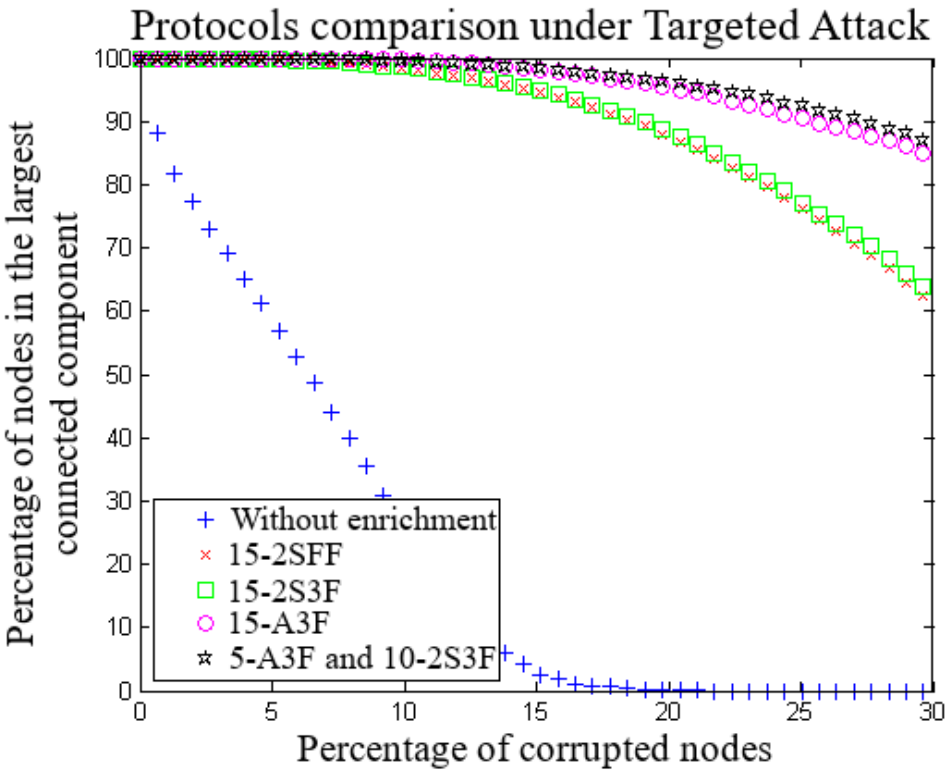


Figure 3.17: Comparison of all protocols for $k = 15$.

Chapter 4

Extending Noiseless Privacy

The standard differential privacy has an obvious drawback which is a necessity of adding a carefully calibrated noise to the final answer to the query. This approach is not always satisfactory, as in some cases we may need to have the **exact** aggregated statistic. Moreover, adding noise, especially individually in distributed case, may lead to significant errors in the aggregated statistic. Finally, adding noise, specifically from a non-standard distribution, can be technically problematic – especially when the aggregated data may come from small, computationally constrained devices. These facts lead to a somewhat reluctant adaptation of the differential privacy notion in real life applications, despite its undeniable merits.
In this chapter we consider relaxation of differential privacy previously presented in [11] called *noiseless privacy*. The intuition behind the *noiseless privacy* approach is that in real life scenarios it might be too pessimistic to assume that the Adversary knows almost every record in the database. This assumption seems far too strong, yet it stands at the heart of standard differential privacy. Indeed, revealing the exact average worldwide income should not do any harm to privacy of any single individual. However, according to differential privacy definition, that would be unacceptable. Intuitively we realize that if an average income (or other value) of a "large" set of participants is revealed, it does not automatically mean that there was a privacy breach understood in a practical, common sense way. These intuitions have already been considered in a few papers, namely [8, 11, 43] to mention the most significant ones, where the authors propose relaxations of the differential privacy model. These relaxations utilize the randomness in the data, which can either come inherently from the data itself, or model the uncertainty of the Adversary. This is contrary to differential privacy which assumes that the Adversary colludes with every other participant.

Unfortunately, previous results are mostly only asymptotic which makes it hard to use in practice, due to unknown constants which may hide the real size of privacy parameters. On the other hand, we focused on detailed, non-asymptotic analysis of the relaxed model to give **explicit bounds** for privacy parameters. Moreover, for the few non-asymptotic results in [11] we show that our methods give better bound for privacy parameters. Furthermore, we also give results for data with (limited) **dependencies**. We want to emphasize that we present the noiseless privacy model in a slightly different way, which seems to be simpler and more convenient.

We focus on extending the types of data which have good noiseless privacy parameters, on introducing dependencies in the data and combining noiseless privacy with standard approach. Moreover, we present detailed results which can be easily applied in real-life scenarios of data aggregation. One could use the notion of noiseless privacy, especially the explicit results given in this chapter, to get rid of, or at least decrease, the noise in privacy preserving data aggregation protocols. To the best of our knowledge, the idea of combining standard differential privacy techniques with adversarial uncertainty was not explored before. Intuitively we can think that in the case where the data has much randomness, we should be able to add less noise than in the case where the data is deterministic from the Adversary's perspective. We give explicit bounds for privacy parameters which allows us to explore the synergy between differential privacy methods and noiseless privacy approach. We describe and analyse this synergy in Section 4.6.

Our results

- We extend the paradigm of utilizing adversarial uncertainty for the case of dependent data (Theorems 13 and 15).
- We explore the synergy between standard differential privacy methods and noiseless privacy approach (Theorem 16).
- We propose an adversarial model (Subsection 4.1.2) and explicit procedure for preserving privacy (Figure 4.6).
- We give improved and explicit (non-asymptotic) bounds for the privacy parameters (Theorems 12 and 14).

We believe that our contribution is a step towards more practical constructions of privacy protocols which utilize adversarial uncertainty. Note that, for the first time, we consider a wide class of dependent data. Moreover, our results state that

the party responsible for privacy does not need to know neither the exact structure of dependencies nor the exact distribution of the data (i.e. joint distribution). Upper bounds for the size of the greatest dependent subset and the sum of centralised third moments (or fourth in case of dependent data) are sufficient to use our results in practice. To achieve it, we used different methods than those used in the context of adversarial uncertainty before.

In Section 4.1 we explain the motivations, recall the idea of utilizing adversarial uncertainty from [11] in a way that is more convenient for presenting our results and provide some formalism that can be seen as an extension of differential privacy notion. We also introduce and discuss our adversarial model and some possible applications. In the next sections we present our results. In Section 4.3 we focus on the case where from the Adversary's perspective the aggregated data is a set of independent random values. Most important is the case discussed in Section 4.4, where we allow the Adversary to know *a priori* some dependencies between data. Note however, that the data owner do not have to know the exact dependencies in the data. Then in Section 4.5 we discuss situation where the Adversary has an exact knowledge of the values of some subset of data values. Finally in Section 4.6 we explore the idea of combining adversarial uncertainty with standard differential privacy approach.

4.1 Model

In the system there are n *users* that may represent different types of parties (organizations, individuals or even sensing devices). Each of them holds a data record x_i (for simplicity we assume that it is a single value). The goal is to aggregate the data and reveal some statistics (say, sum of the values). Note that the database may either be a centralized one or a distributed one where users themselves have to generate some output according to a distributed protocol. See that in terms of privacy definition, both these cases are equivalent. They differ in algorithmic approach to these problems and Adversary's capabilities. Here we introduce some notation

- *data* - the set of n values (held either by different parties or by a single curator) which we want to aggregate (e.g. compute the sum of these values) and reveal the obtained statistic to the public,
- *compromised users* - the subset of data about which the adversary has full knowledge, namely he knows the exact values in this subset,

- *data owner* - the party that is responsible for preserving privacy of the data by designing an appropriate algorithm, choosing protocol parameters according to the expected power of the Adversary (or upper bounds for them) or deciding whether specific privacy parameters are sufficient or if they have to be combined with external noise.

4.1.1 Modeling Privacy of Randomized Data

We use a privacy model in which the data (or at least part of it) is considered random from the Adversary's perspective, coming from a specific distribution. This kind of approach is quite natural in many scenarios, namely the Adversary's knowledge is usually limited. However, it needs a different definition of privacy than standard differential privacy as in [28], because we have to take into account randomized inputs. Following the notion introduced in [11] we call this approach *noiseless privacy*. Before we show its formal definition, we need to introduce the following

Definition 24 (Adjacent Random Vectors)**.** Let $X = (X_1, \ldots, X_n)$ be an arbitrary random vector and let X' be other random vector. Let X_* be a random variable. We will say that vectors X and X' are adjacent if and only if

$$X' = (X_1, \ldots, X_i, X_*, X_{i+1}, \ldots, X_n),$$

or

$$X' = (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n),$$

for any $i \in \{1, \ldots, n\}$.

This essentially captures the notion of data vectors adjacency similar to the one in [28], but for random variables rather than deterministic values. See that if for some deterministic adjacent vectors x and x' we have $X = x$ and $X' = x'$ with probability 1, then this definition of adjacency is the same as in [28]. Note that we could as well define adjacency in such a way that instead of adding or removing a vector element, we could simply change its value. This is just the matter of choice and a few straightforward technical changes in proofs. Continuing, we can introduce the following

Definition 25 (Data sensitivity)**.** We will say that data vector $X = (X_1, \ldots, X_n)$ and mechanism M have data sensitivity Δ if an only if

$$|M(X) - M(X')| \leqslant \Delta,$$

almost surely for every vector X' that is adjacent to X.

Note that this bears close resemblance to the l_1-sensitivity defined in [28]. More detailed comparison of noiseless privacy and standard differential privacy can be found in Section 4.2.

We can formally define noiseless privacy in the following way

Definition 26 (Noiseless Privacy)**.** We say that a privacy mechanism M for a random vector $X = (X_1, \ldots, X_n)$ preserves noiseless privacy with parameters (ε, δ) if for all $\mathcal{B} \subseteq \text{Range}(M)$ and any random vector X' such that X and X' are adjacent we have

$$\mathbb{P}(M(X) \in B) \leqslant e^{\varepsilon}\mathbb{P}(M(X') \in B) + \delta.$$

Intuitively, this definition says that if data can be considered random, then the outcome of the coin flip of any single user does not significantly change the result of **deterministic** mechanism M, whether the user is added to the result, or removed from it. This is very similar to standard differential privacy, however here the data itself is considered randomized and therefore impacting the privacy parameters. We will use abbreviation (ε, δ)-NP to denote noiseless privacy with parameters ε and δ.

Clearly, this model of privacy is a coherent extension of differential privacy. We see it as a generalization of the known differential privacy definition that can be useful for some real life scenarios. See that in Remark 3 we explained that this model is indeed more general than differential privacy, but if we fix the data as deterministic, it is essentially the same definition. Moreover, in Section 4.6 we show how the standard differential privacy methods can be combined with noiseless privacy approach.

Whether or not (and to what extent) particular data can be considered random is of course an important problem to be solved by the data holder, but we do not focus on it in our research. Note that also other papers in this line of research has not yet dealt with this problem which may be a very interesting question for future work.

See that in noiseless privacy, random data has natural self-hiding properties, even though the mechanisms are deterministic. Instead of relying on the randomness of mechanism (as in the standard differential privacy methods), we can sometimes rely on the inherent randomness of the data itself. Deterministic algorithms have an obvious benefit of not introducing any errors (which are inevitable in standard differential privacy approach due to the randomness introduced), so the answer to a query is exact.

The most common and useful deterministic mechanism would be simply summing all the data. We explore the privacy parameters of mechanism $M(X) = sum(X)$ for any distribution of the data vector X, a wide class of dependencies in the data and the adversarial model defined in Subsection 4.1.2.

4.1.2 Adversarial Model

We assume that the Adversary:

- may know the exact data of at most some fraction $0 \leqslant \gamma < 1$ of the users,
- may know the correct distribution (but not the value itself) of the data of the rest of users (note that the distribution for each user might be different),
- may know the dependencies between some of the data values (if there are any), but only in subsets of size at most D.

Let us now discuss and justify these assumptions. First of all, one can easily see that in standard differential privacy we essentially assume that the Adversary knows the exact data of all users except one. Here we relax this by giving an upper bound on the number of users which are compromised. See that in realistic scenarios it is not very plausible that the Adversary indeed knows almost every data record. On the other hand, we still give him quite a lot of power, namely we assume that he knows the distributions of the data, but not the exact values. From the point of view of the Adversary, data is a vector of (at least $n - \gamma n$) random variables with known distribution and some known (at most γn) data values. See that in sections 4.3 and 4.4 we assume for simplicity that the Adversary does not know any exact values (so $\gamma = 0$). We discuss this in Section 4.5 where we show how to extend our results for the case where the Adversary knows any arbitrary γn exact values.

In real-life data it is quite common to have some dependencies involved. Moreover, the Adversary might know about them. To propose a realistic model for noiseless privacy, one has to take it into account. In our model we give the Adversary the precise knowledge about all dependencies in subsets of size at most D. That essentially means that he does not have an insight into dependencies of subsets of size greater than D. Note that it might be the case that such dependencies do not exist (the data might really have all dependent subsets of size at most D), or simply the Adversary does not know about these dependencies and cannot utilize them. Obviously in standard differential privacy notion we do not care about

the distribution of data, whether it is dependent or not. Here, on the other hand, due to the necessity of utilizing the inherent randomness in data instead of adding external noises, we must take it into account.

See that there is an asymmetry between the Adversary and other users and even the data owner. We assume that the Adversary has power of knowing the exact dependencies (of size at most D), while neither users nor the data owner have to know neither the dependencies nor the joint distribution of the data. The parameter necessary to use our results is the upper bound for D. Note that the data owner might do some tests for independence of the data (or subsets of the data), using standard statistical methods for testing independence, e.g. χ^2-test. Information about the upper bound for the size of dependent subsets might also come from strictly engineering knowledge, say due to physical proximity of the subset of sensors or some social knowledge, say subset of users having the same age. This approach to dependencies essentially boils down to the known notion of *dependency neighborhoods* defined in Subsection 1.4.3.

Observe that the definition of dependency neighborhoods actually says that for specific X_i we know that it is independent of those that are not in its neighborhood. We want to give a general approach to local dependencies scenario, so in our analysis we do not assume anything about joint distributions of the dependent subsets (the dependency in subset might even mean 'equality') .

To sum it up, we present a formal definition of adversarial model.

Definition 27. We will denote a specific instantiation of adversarial model for data vector X by $Adv_X(D, \gamma)$, where

- D is an upper bound for the size of the greatest dependent subset,
- γ is the upper bound for the fraction of the data which values the Adversary exactly knows,
- the Adversary knows the distribution of data vector X.

We believe that while our adversarial model gives significantly less power to the Adversary than in standard differential privacy notion (which basically gives the Adversary almost full knowledge of the data), they still are reasonable and applicable in real-life scenarios. One important remark is that we **do not** need to predict the exact Adversary's knowledge about the dependencies. We only need to know the maximum size of dependency neighborhood, namely the size of largest non-independent subset of data. In fact, we only need an **upper bound** for that

size. Same with the fraction of data values which the Adversary knows. To apply our results, which are presented in the next sections, one will also need a lower bound for the variance of data and upper bound for the sum of third and fourth centralized moments for the specific data vector.

4.2 Comparison to Standard Differential Privacy

Clearly noiseless privacy is an extension of the standard differential privacy that is applicable to the case when we can assume that the observer/attacker may treat the raw data of users (before being processed) as random variables. In particular if we assume that all data items are concentrated in single points (i.e, $\mathbb{P}(X_i = x_i) = 1$ for all i) we get the original (ε, δ)-differential privacy.

While the standard differential privacy definition guarantees immunity against attacks based on *auxiliary information* (e.g. from publicly available datasets or even personal knowledge about an individual participating in the protocol), the noiseless privacy is more general as we can either assume that the adversary has no auxiliary information, or assume that there is an upper bound on the size of subset of database entries about which he has some external knowledge. Note that if we assume full auxiliary information noiseless privacy becomes completely unacceptable, which is very intuitive, as the whole notion of adversarial uncertainty demands that the adversary does not have full knowledge. Moreover, it is often too pessimistic to assume that the adversary knows everything except for the single data record which privacy he wants to breach.

Remark 3. See that in the standard differential privacy definition (e.g. [28]) we essentially want

$$\mathbb{P}(M(X) \in B | X = x) \leqslant e^{\varepsilon} \mathbb{P}(M(X') \in B | X' = x') + \delta,$$

where x and x' are adjacent, deterministic vectors.
This captures the notion of neighboring databases. Our approach is indeed a relaxation of that definition, as we do not necessarily condition the data to have some fixed, deterministic value. We treat the data inputs as random variables. In particular, if we have $X = x$ with probability 1 then our model collapses to standard differential privacy.

Differential privacy has some very useful properties. First of all, it is immune to post-processing, so the adversary cannot get any additional information,

and consequently cannot increase the privacy loss by convoluting the result of a mechanism with some deterministic function.

Noiseless privacy is, similarly to standard differential privacy, resilient to post-processing.

Fact 7. *Let privacy mechanism M and a random vector X be such that $M(X)$ is (ε, δ)-NP. Let $f : R \to R'$ be an arbitrary randomized mapping. Then $f(M(X))$ is also (ε, δ)-NP.*

Proof As in [28] we prove this fact for a deterministic function $f : R \to R'$. The result follows because any randomized mapping can be decomposed into a convex combination of deterministic functions. Such combination of noiselessly private functions is also noiselessly private. Fix any adjacent pair of vectors X and X'. Let $T = \{r \in R : f(r) \in S\}$ and fix $S \subset R'$. We have

$$\begin{aligned}&\mathbb{P}(f(M(X)) \in S) = \mathbb{P}(M(X) \in T) \leqslant \\ &\leqslant e^{\varepsilon}\mathbb{P}(M(X') \in T) + \delta = e^{\varepsilon}\mathbb{P}(f(M(X')) \in S) + \delta.\end{aligned}$$

□

Another important property of differential privacy is its composability. There has been an extended discussion concerning composability of noiseless privacy and its derivatives in [8, 11, 43].

4.3 Explicit Bounds for Independent Data

In this section we consider noiseless privacy for independent data. We aim to show explicit bounds for general case. Throughout this section we assume that we have a database X which consists of n values so $X = (X_1, \ldots, X_n)$.

4.3.1 Binomially Distributed Data

Let us consider a simple, warm-up scenario, where X_i are i.i.d. random variables and $X_i \sim Bin(1, p)$. We want to aggregate the sum of all these variables so we set the mechanism as $M(X) = \sum_{i=1}^{n} X_i \sim Bin(n, p)$. First, let us prove a useful lemma

Lemma 6. *Let $M \sim Bin(n,p)$. Fix an arbitrary $\lambda > 0$ such that $(np-\lambda) > 0$ and $(np+\lambda) < n$. Let $u \in [np-\lambda, np+\lambda] \cap \mathbb{Z}$ and let $v \in \mathbb{Z}$ such that $|u-v| = 1$. Then for*

$$\varepsilon = \begin{cases} \frac{\lambda}{n}\left(\frac{1+\frac{1}{\lambda}}{1-p} - \frac{1}{\frac{\lambda}{n}-p}\right), p \leqslant \frac{1}{2}, \\ \frac{\lambda}{n}\left(\frac{1+\frac{1}{\lambda}}{p} - \frac{1}{\frac{\lambda}{n}-(1-p)}\right), p > \frac{1}{2}, \end{cases}$$

we have

$$\mathbb{P}(M = u) \leqslant e^{\varepsilon}\mathbb{P}(M = v).$$

Proof We want to bound $\frac{\mathbb{P}(M=u)}{\mathbb{P}(M=v)}$, where $|u-v| = 1$ and $M \sim Bin(n,p)$. Furthermore, we know that $u \in [np-\lambda, np+\lambda] \cap \mathbb{Z}$. First observe that we get the biggest ratio either for the smallest or greatest possible u. If $p \leqslant \frac{1}{2}$ we get the biggest ratio for the smallest possible u and if $p > \frac{1}{2}$ then we get the biggest ratio for the largest possible u. Therefore it remains to check these two cases, calculate ε_1 and ε_2 and pick $\varepsilon = \max(\varepsilon_1, \varepsilon_2)$.

Let us begin with the case where $p \leqslant \frac{1}{2}$. One can easily check that the greatest possible ratio is for $u = \lceil np-\lambda \rceil$ and $v = (u-1)$. We can bound it in the following way

$$\begin{aligned} \frac{\mathbb{P}(M = \lceil np-\lambda \rceil)}{\mathbb{P}(M = \lceil np-\lambda \rceil - 1)} &= \frac{n - \lceil np-\lambda \rceil + 1}{\lceil np-\lambda \rceil} \cdot \frac{p}{1-p} \leqslant \\ &\leqslant \frac{n-np+\lambda+1}{np-\lambda} \cdot \frac{p}{1-p} = \\ &= \frac{1+\frac{\lambda+1}{n(1-p)}}{1-\frac{\lambda}{np}} \leqslant \frac{\exp(\frac{\lambda+1}{n(1-p)})}{1-\frac{\lambda}{np}}. \end{aligned}$$

Ultimately we are interested in the natural logarithm of that ratio. We have

$$\begin{aligned} \varepsilon_1 &= \ln\left(\frac{\exp(\frac{\lambda+1}{n(1-p)})}{1-\frac{\lambda}{np}}\right) = \frac{\lambda+1}{n(1-p)} - \ln\left(1-\frac{\lambda}{np}\right) \leqslant \\ &\leqslant \frac{\lambda+1}{n(1-p)} - 1 + \frac{1}{1-\frac{\lambda}{np}} = \lambda\left(\frac{1+\frac{1}{\lambda}}{n(1-p)} + \frac{1}{np-\lambda}\right) = \\ &= \frac{\lambda}{n}\left(\frac{1+\frac{1}{\lambda}}{1-p} - \frac{1}{\frac{\lambda}{n}-p}\right), \end{aligned}$$

where the inequality comes from the fact that $(1-\frac{1}{x}) \leqslant \ln(x)$ for $x > 0$. See also that $1-\frac{\lambda}{np} > 0$, because we assumed that $(np-\lambda) > 0$. Note that we picked the

biggest possible ratio, so for $p \leqslant \frac{1}{2}$ it is true for every $u \in [np - \lambda, np + \lambda] \cap \mathbb{Z}$ that

$$\frac{\mathbb{P}(X = u)}{\mathbb{P}(X = v)} \leqslant e^{\varepsilon_1} \iff \mathbb{P}(X = u) \leqslant e^{\varepsilon_1}\mathbb{P}(X = v),$$

where $|u - v| = 1$. Now let us assume that $p > \frac{1}{2}$. In that case the greatest possible ratio is for $u = (np+\lambda)$ and $v = (u+1)$. One can easily see, that we can simply consider $Bin(n, 1 - p)$ and apply exactly the same reasoning as before. That leaves us with

$$\varepsilon_2 = \frac{\lambda}{n}\left(\frac{1 + \frac{1}{\lambda}}{p} - \frac{1}{\frac{\lambda}{n} - (1 - p)}\right).$$

We conclude that for a fixed λ we have the following:

$$\varepsilon = \begin{cases} \frac{\lambda}{n}\left(\frac{1+\frac{1}{\lambda}}{1-p} - \frac{1}{\frac{\lambda}{n}-p}\right), p \leqslant \frac{1}{2}, \\ \frac{\lambda}{n}\left(\frac{1}{p} - \frac{1+\frac{1}{\lambda}}{\frac{\lambda}{n}-(1-p)}\right), p > \frac{1}{2}. \end{cases}$$

In the end we found ε, such that for all $u \in [np - \lambda, np + \lambda] \cap \mathbb{Z}$ and $|u - v| = 1$ it holds that

$$\mathbb{P}(X = u) \leqslant e^{\varepsilon}\mathbb{P}(X = v),$$

which concludes the proof of this lemma. □

Now we can state a theorem which shows that i.i.d. binomial data has very strong noiseless privacy properties for a wide range of parameters.

Theorem 11. *Let $X = (X_1, \ldots, X_n)$ where $X_i \sim Bin(1, p)$ are i.i.d. random variables. Let $M(X) = \sum_{i=1}^{n} X_i$ and fix $\delta \in (0, 1)$. Moreover, let us assume that $p \in \left(\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}, 1 - \sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}\right)$. Then for*

$$\varepsilon = \begin{cases} \sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}\left(\frac{1+\sqrt{\frac{2}{n\ln\frac{2}{\delta}}}}{1-p} - \frac{1}{\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}-p}\right), p \leqslant \frac{1}{2}, \\ \sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}\left(\frac{1+\sqrt{\frac{2}{n\ln\frac{2}{\delta}}}}{p} - \frac{1}{\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}-(1-p)}\right), p > \frac{1}{2}, \end{cases}$$

$M(X)$ is (ε, δ)-NP. On the other hand, if $\varepsilon > 0$ is fixed, then for

$$\delta = \begin{cases} 2\exp\left(-2np^2\left(1 - \frac{1}{e^{\varepsilon}(1-p)+p}\right)^2\right), p \leqslant \frac{1}{2}, \\ 2\exp\left(-2n(1-p)^2\left(1 - \frac{1}{e^{\varepsilon}p+(1-p)}\right)^2\right), p > \frac{1}{2}, \end{cases}$$

$M(X)$ is (ε, δ)-NP.

Proof Let us begin with the first case, where δ is fixed. One obvious observation is that $M(X) \sim Bin(n, p)$. Let $\lambda = \sqrt{\frac{n \ln \frac{2}{\delta}}{2}}$. Note that $\lambda > 0$, $np + \lambda < n$ and $np - \lambda > 0$. Using Chernoff bound (see for example [22]) for binomial distribution we get

$$\mathbb{P}(M(X) \geqslant np + \lambda) + \mathbb{P}(M(X) \leqslant np - \lambda) \leqslant 2\exp\left(-\frac{2\lambda^2}{n}\right) = \delta,$$

as we want to limit the tail probability by parameter δ. Let us denote the set $S = \{\lceil np - \lambda \rceil, \ldots, \lfloor np + \lambda \rfloor\}$, which is exactly the support of $M(X)$ without the tails which probability we just bounded by δ. Now we have to show that, apart from these tails, for given ε the following holds

$$\forall_{B \subset S}\left(\left|\ln\left(\frac{\mathbb{P}(M(X) \in B)}{\mathbb{P}(M(X') \in B)}\right)\right| \leqslant \varepsilon\right).$$

It is easy to see that instead of considering all subsets of S, we can consider only the single values, because taking a single value with a bigger ratio yields worst case bound. For that we can use Lemma 6. We indeed have $M(X) \sim Bin(n, p)$. Moreover, we have $\lambda = \sqrt{\frac{n \ln \frac{2}{\delta}}{2}}$ so $(np - \lambda) > 0$ and $(np + \lambda) < n$. See also that for $X_i \sim Bin(1, p)$ i.i.d. we have data sensitivity 1. One can easily see that adding or removing a single data point can change the sum at most by 1, therefore we consider u and v such that $|u - v| = 1$. Observe that $\frac{\lambda}{n} = \sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}$, therefore we have

$$\varepsilon = \begin{cases} \frac{\lambda}{n}\left(\frac{1+\frac{1}{\lambda}}{1-p} - \frac{1}{\frac{\lambda}{n}-p}\right), p \leqslant \frac{1}{2}, \\ \frac{\lambda}{n}\left(\frac{1+\frac{1}{\lambda}}{p} - \frac{1}{\frac{\lambda}{n}-(1-p)}\right), p > \frac{1}{2}. \end{cases}$$

Applying Lemma 6 for $M(X)$, λ and ε we obtain that

$$\mathbb{P}(M(X) = u) \leqslant e^{\varepsilon}\mathbb{P}(M(X) = v),$$

for $u \in S$ and $|u - v| \leqslant 1$. Therefore $M(X)$ is $(\epsilon, \delta) - NP$, namely we have

$$\mathbb{P}(M(X) \in S) \leqslant e^{\varepsilon}\mathbb{P}(M(X') \in S) + \delta,$$

where X and X' are adjacent vectors and $\varepsilon = \varepsilon(n, p, \delta)$. The addition of δ comes from the fact that we bound the tails of $M(X)$.

Now we assume that we have a fixed $\varepsilon > 0$. Let $\alpha = e^{\varepsilon}$ and $w = \frac{p}{1-p}$. We use similar reasoning as in Lemma 6. First let us consider $p \leqslant \frac{1}{2}$. We are interested in the greatest integer k smaller than np, which does **not** satisfy the following

$$\frac{\mathbb{P}(M(X) = k)}{\mathbb{P}(M(X) = k - 1)} \leqslant \alpha.$$

We have

$$\frac{\mathbb{P}(M(X) = k)}{\mathbb{P}(M(X) = k - 1)} = \frac{n - k + 1}{k} \cdot w > \alpha \iff k < \frac{w(n+1)}{\alpha + w}.$$

Now let us pick $\lambda_k = \mu - k > \mu - \frac{(n+1)w}{\alpha+w}$ where $\mu = np$. We will bound the tail using Chernoff bound

$$\begin{aligned}
P(M(X) \leqslant \mu - \lambda_k) &\leqslant \exp\left(\frac{-2\lambda_k^2}{n}\right) < \\
&< \exp\left(\frac{-2\left(\mu - \frac{(n+1)w}{\alpha+w}\right)^2}{n}\right) = \\
&= \exp\left(-2np^2\left(1 - \frac{n+1}{n} \cdot \frac{1}{\alpha(1-p)+p}\right)^2\right) < \\
&< \exp\left(-2np^2\left(1 - \frac{1}{\alpha(1-p)+p}\right)^2\right).
\end{aligned}$$

Now see that

$$\begin{aligned}
&\mathbb{P}(M(X) \leqslant \mu - \lambda_k) + \mathbb{P}(M(X) \geqslant \mu + \lambda_k) \leqslant \\
&\leqslant 2\exp\left(-2np^2\left(1 - \frac{1}{e^{\varepsilon}(1-p)+p}\right)^2\right) = \delta.
\end{aligned}$$

When $p > \frac{1}{2}$ we can do similar symmetric reasoning as before, we obtain

$$\mathbb{P}(M(X) \leqslant \mu - \lambda_{k'}) + \mathbb{P}(M(X) \geqslant \mu + \lambda_{k'}) \leqslant$$
$$\leqslant 2\exp\left(-2n(1-p)^2\left(1 - \frac{1}{e^{\varepsilon}p + (1-p)}\right)^2\right) = \delta,$$

where k' is the smallest integer larger than np which does **not** satisfy the following

$$\frac{\mathbb{P}(M(X) = k')}{\mathbb{P}(M(X) = k' + 1)} \leqslant \alpha.$$

This concludes the proof, because we bounded the subset of possible values which did not satisfy our required ratio. In the end we have

$$\mathbb{P}(M(X) \in S) \leqslant e^{\varepsilon}\mathbb{P}(M(X') \in S) + \delta.$$

□

Let us observe that in Theorem 11 for constant parameters p and δ we get $\varepsilon = O\left(\frac{1}{\sqrt{n}}\right)$. It is also worth noting that for p close to $\sqrt{\frac{\ln\frac{2}{\delta}}{2n}}$ or $1 - \sqrt{\frac{\ln\frac{2}{\delta}}{2n}}$, ε can be large, although as long as p is constant, ε still approaches 0 with $n \to \infty$.

Similarly, for p very close to 0 or 1 and for small n, the value of δ can be large. Nevertheless we see that δ is decreasing **exponentially** to 0 with $n \to \infty$, so for sufficiently large n we still get very small values of δ.

As one can see in figures 4.1 and 4.2, our Theorem does not only give non-asymptotical, explicit parameters (both for the case where ε is fixed and the case where δ is fixed), but also, due to slightly more careful reasoning, our bound is tighter than the bound which authors of [11] have implicitly shown in their proof.

That was just a warm-up scenario to show how does noiseless privacy work with simple data distribution. Let us move to a more interesting model where users' data has different, but still independent distributions.

4.3.2 General Case

From now on we do not assume any specific distribution of the data. First let us recall Theorem 5 from Subsection 1.6.3. See that it can also be stated in the following way, which will be useful for us throughout this subsection

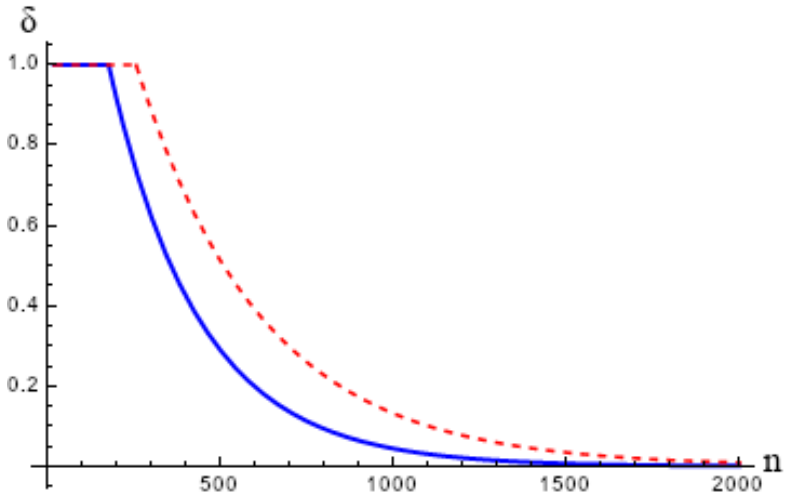


Figure 4.1: $\varepsilon = 0.5$, $p = 0.95$, red dashed line is guarantee for parameter δ in paper [11], blue thick line is guarantee from our Theorem 11.

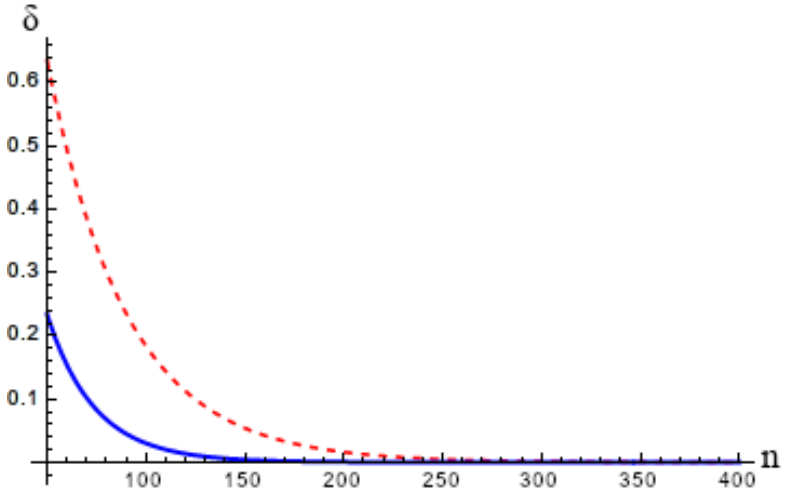


Figure 4.2: $\varepsilon = 1$, $p = 0.2$, red dashed line is guarantee for parameter δ in paper [11], blue thick line is guarantee from our Theorem 11.

Corollary 5. *Fix $\varepsilon \in (0,1)$ and $\delta > 0$ and denote the data sensitivity Δ. For random variable $Z \sim \mathcal{N}(0, \sigma^2)$, where $\sigma > \frac{\Delta\sqrt{2\ln(\frac{1.25}{\delta})}}{\varepsilon}$ we have*

$$P[u + Z \in S] \leqslant e^{\varepsilon} P[v + Z \in S] + \delta,$$

where u and v are any real numbers such that $|u - v| \leqslant \Delta$.

Now we present the general theorem for independent data

Theorem 12. *Let $X = (X_1, \dots, X_n)$ where X_i's are independent random variables. Let $\mu_i = \mathbb{E}X_i$, $\sigma_i^2 = \mathbb{V}ar(X_i)$, $\sigma^2 = \frac{\sum_{i=1}^n \sigma_i^2}{n}$ and $\mathbb{E}|X_i|^3 < \infty$ for every $i \in [n]$. Let $M(X) = \sum_{i=1}^n (X_i)$. Assume that $\sqrt{\frac{\Delta^2 \ln(n)}{n\sigma^2}} < 1$, where Δ is the data sensitivity of $M(X)$. Then for ε such that*

$$\sqrt{\frac{\Delta^2 \ln(n)}{n\sigma^2}} < \varepsilon < 1,$$

and

$$\delta = \frac{1.12 \sum_{i=1}^n \mathbb{E}|X_i - \mu_i|^3}{(n\sigma^2)^{\frac{3}{2}}}(1 + e^{\varepsilon}) + \frac{5}{4\sqrt{n}},$$

Proof To prove this theorem, we use Corollary 5 and Fact 2 from Subsection 1.4.3. Let $u, v \in supp(M(X))$ and let $|u - v| \leqslant \Delta$. For any set B we can denote $B_u = \{b + u : b \in B\}$. For simplicity let us, for now, assume that $\mu_i = 0$ for every i. From assumptions we also know that $\mathbb{E}|X_i|^3 < \infty$ for every i, so we can use Fact 2. Let $Z \sim \mathcal{N}(0, n\sigma^2)$. For every B_u we have

$$\mathbb{P}\left(M(X) \in B_u\right) \leqslant \mathbb{P}\left(Z \in B_u\right) + 2d_k(M(X), Z) \leqslant \mathbb{P}\left(Z \in B_u\right) + 2\delta_1,$$

where δ_1 is the rate of convergence described in Fact 2. Recall that

$$\sqrt{\frac{\Delta^2 \ln(n)}{n\sigma^2}} < \varepsilon < 1,$$

and let $\delta_2 = \frac{5}{4\sqrt{n}}$. See that

$$\sqrt{n}\sigma > \frac{\Delta\sqrt{2\ln(\frac{1.25}{\delta_2})}}{\varepsilon} = \frac{\Delta\sqrt{\ln n}}{\varepsilon} \iff \varepsilon > \sqrt{\frac{\Delta^2 \cdot \ln n}{n\sigma^2}}.$$

See that $Z \sim \mathcal{N}(0, n\sigma^2)$ and $|u - v| \leqslant \Delta$ so we can use Corollary 5:

$$\mathbb{P}\left(Z \in B_u\right) + 2\delta_1 \leqslant e^{\varepsilon}\mathbb{P}\left(Z \in B_v\right) + 2\delta_1 + \delta_2.$$

Now we have to return to our initial distribution. Again, we use Fact 2.

$$e^{\varepsilon}\mathbb{P}\left(Z \in B_v\right) + 2\delta_1 + \delta_2 \leqslant e^{\varepsilon}\mathbb{P}\left(M(X) \in B_v\right) + 2\delta_1(1 + e^{\varepsilon}) + \delta_2.$$

Note that for simplicity we assumed $\mathbb{E}X_i = 0$. See that for $Y_i = (X_i - \mu_i)$, where $\mu_i = \mathbb{E}X_i$ the proof is still correct. Recall from Fact 2 that $\delta_1 \leqslant \frac{0.56 \sum_{i=1}^{n} \mathbb{E}|X_i - \mu_i|^3}{\left(\sum_{i=1}^{n} \sigma_i^2\right)^{\frac{3}{2}}}$. Now see that

$$2\delta_1(1 + e^{\varepsilon}) + \delta_2 \leqslant \frac{1.12 \sum_{i=1}^{n} \mathbb{E}|X_i - \mu_i|^3}{\left(\sum_{i=1}^{n} \sigma_i^2\right)^{\frac{3}{2}}}(1 + e^{\varepsilon}) + \frac{5}{4\sqrt{n}} = \delta.$$

Therefore

$$\mathbb{P}(M(X) \in B_u) \leqslant e^{\varepsilon}\mathbb{P}(M(X) \in B_v) + \delta,$$

which concludes that $M(X)$ is (ε, δ)-NP. □

Theorem 12 gives us very general notion of privacy parameters for summing independent data. Note that we assumed nothing about the distribution of the data, apart from being independent. The only values we need to know is the variance and sum of appropriate central moments (or upper bounds for these values). Data independence is obviously a strong (and generally false) assumption in real world, but it is commonly used. However, we will also work with **dependent** data in the next section. We also present an example.

Example 5. We consider a data vector $X = (X_1, \ldots, X_n)$, where X_i's are independent random variables. Let $\Delta = 30$. Let $\sigma^2 = \frac{\sum_{i=1}^{n} \sigma_i^2}{n} = 4$. Let also $\sum_{i=1}^{n} \mathbb{E}|X_i - \mu_i|^3 = 3 \cdot n$. We use mechanism $M(X) = \sum_{i=1}^{n} X_i$. Using Theorem 12 we obtain that it is (ε, δ)-NP when $\varepsilon < 1$. Figure 4.3 shows how the ε decreases with n, while Figure 4.4 shows how δ decreases with n. Note that the conditions of Theorem 12 are satisfied from n approximately the size of 2000 in this case.

We can see that for n around 10000 parameter δ is smaller than 0.05, which is a constant widely used in differential privacy literature, and decreases further. Also, note that for $n \geqslant 10000$ the parameter ε is below 0.5 which also is a widely used constant in differential privacy papers (see for example [17]). Clearly, the parameters keep improving with more users.

4.4 Explicit Bounds for Locally Dependent Data

In the previous section we gave a general treatment for privacy parameters of independent variables. However, in many cases the data has some local dependencies

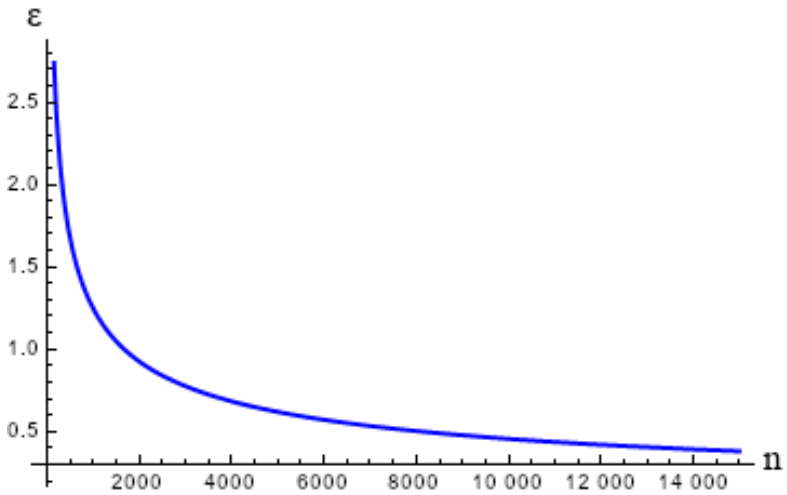


Figure 4.3: Parameter ε in Example 5.

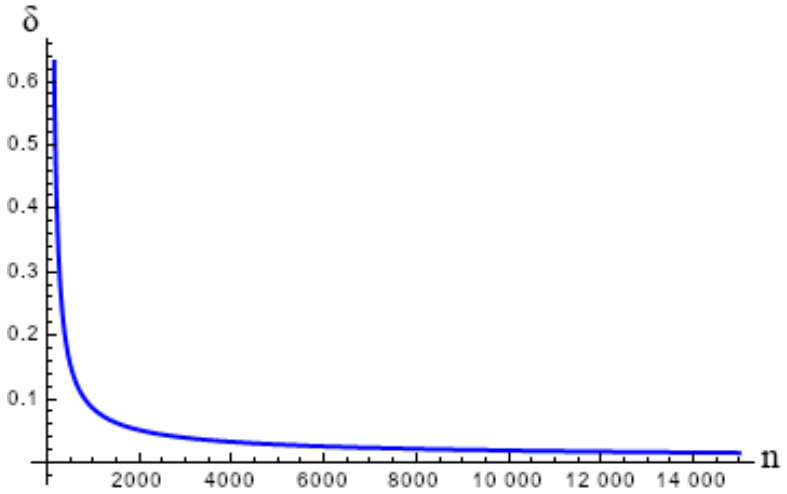


Figure 4.4: Parameter δ in Example 5.

involved. Imagine a situation where we want to collect the data of yearly salary from former students of a specific university. Say, those that finished their education at most 5 years ago. Our goal is to obtain the average yearly salary of all students that finished their education during last five years. Now one can easily see that there will be some local dependencies between the participants as some of the students might work in the same company, launch a startup together or just

work in the same field. This will affect their salary and therefore make it locally dependent. Such dependencies are modeled using *dependency neighborhoods* notion, which we defined in Subsection 1.4.3.

As previously, we want to take the sum of all our data and show privacy parameters for this mechanism. We are going to take a similar approach as in Theorem 12. That is, we want to bound the distance between the distribution of sum of our data and normal distribution. Then, using standard differential privacy properties of normal distribution (described in Corollary 5) we derive privacy parameters. However, this time we cannot use Berry-Esseen theorem to bound the mentioned distance, as the data is not independent. Instead, we use Stein's method (see Subsection 1.4.3), which allows to bound the Kolmogorov distance between two random variables. Apart from that, the presented reasoning is very similar to the one Theorem 12.

Theorem 13. *Let $X = (X_1, \ldots, X_n)$, where X_i's are, possibly dependent, random variables and $M(X) = \sum_{i=1}^{n}(X_i)$. Let $\mathbb{E}X_i = \mu_i$ and $\mathbb{E}X_i^4 < \infty$. Assume there are dependency neighborhoods N_i, $i \in \{1, \ldots, n\}$, and $D = \max_{1 \leqslant i \leqslant n} |N_i|$. Let $\sigma^2 = \mathbb{V}ar(M(X))$ and data sensitivity of $M(X)$ be Δ. We also assume that $0 < \sqrt{\frac{\Delta^2 \ln(n)}{\sigma^2}} < 1$. Then for ε such that*

$$\sqrt{\frac{\Delta^2 \ln(n)}{\sigma^2}} < \varepsilon < 1,$$

and

$$\delta = c(\varepsilon)\sqrt{\frac{D^2}{\sigma^3}\sum_{i=1}^{n}\mathbb{E}|X_i^*|^3 + \frac{D^{\frac{3}{2}}\sqrt{26}}{\sigma^2\sqrt{\pi}}\sqrt{\sum_{i=1}^{n}\mathbb{E}\left(X_i^*\right)^4}} + \frac{5}{4\sqrt{n}},$$

where $X_i^* = (X_i - \mu_i)$ and

$$c(\varepsilon) = 2(1 + e^{\varepsilon})\left(\frac{2}{\pi}\right)^{\frac{1}{4}},$$

$M(X)$ is (ε, δ)-NP.

Proof To prove this theorem, we use Kolmogorov and Wasserstein distances, which were defined in Subsection 1.4.3 in Definition 4 and Definition 5 and also facts stated in Subsection 1.4.3, namely Fact 3 and Fact 4. Let $u, v \in supp(M(X))$ and $|u - v| \leqslant \Delta$. For set B let us denote $B_u = \{b + u : b \in B\}$.

Moreover, throughout the proof we denote $\frac{B_u}{\sigma} = \{\frac{b}{\sigma} : b \in B_u\}$. For simplicity let us, for now, assume that $\mathbb{E}X_i = 0$ for every i. Let $Y \sim \mathcal{N}(0,1)$ and $Z \sim \mathcal{N}(0,\sigma^2)$. For every B_u we have

$$\mathbb{P}\left(M(X) \in B_u\right) = \mathbb{P}\left(\frac{M(X)}{\sigma} \in \frac{B_u}{\sigma}\right) \leqslant \mathbb{P}\left(Y \in \frac{B_u}{\sigma}\right) + 2d_K\left(\frac{M(X)}{\sigma}, Y\right) =$$
$$= \mathbb{P}\left(Z \in B_u\right) + 2d_K\left(\frac{M(X)}{\sigma}, Y\right).$$

Recall that

$$\sqrt{\frac{\Delta^2 \ln(n)}{\sigma^2}} < \varepsilon < 1,$$

and let $\delta_1 = \frac{5}{4\sqrt{n}}$. See that

$$\sigma > \frac{\Delta\sqrt{2\ln(\frac{1.25}{\delta_2})}}{\varepsilon} = \frac{\Delta\sqrt{\ln n}}{\varepsilon} \iff \varepsilon > \sqrt{\frac{\Delta^2 \cdot \ln n}{\sigma^2}}.$$

Therefore we can use the property of the normal distribution stated in Corollary 5.

$$\mathbb{P}\left(Z \in B_u\right) + 2d_K\left(\frac{M(X)}{\sigma}, Y\right) \leqslant e^{\varepsilon}\mathbb{P}\left(Z \in B_v\right) + 2d_K\left(\frac{M(X)}{\sigma}, Y\right) + \delta_1.$$

Now we have to return to our initial distribution.

$$e^{\varepsilon}\mathbb{P}\left(Z \in B_v\right) + 2d_K\left(\frac{M(X)}{\sigma}, Y\right) + \delta_1 \leqslant$$
$$\leqslant e^{\varepsilon}\mathbb{P}\left(M(X) \in B_v\right) + 2d_K\left(\frac{M(X)}{\sigma}, Y\right)(1 + e^{\varepsilon}) + \delta_1.$$

So far we have

$$\mathbb{P}\left(M(X) \in B_u\right) \leqslant e^{\varepsilon}\mathbb{P}\left(M(X) \in B_v\right) + 2d_K\left(\frac{M(X)}{\sigma}, Y\right)(1 + e^{\varepsilon}) + \delta_1.$$

We use Fact 3 to obtain

$$d_K\left(\frac{M(X)}{\sigma}, Y\right) \leqslant \left(\frac{2}{\pi}\right)^{\frac{1}{4}}\sqrt{d_W\left(\frac{M(X)}{\sigma}, Y\right)}$$

Recall that we assumed $\mathbb{E}X_i = 0$ and $\mathbb{E}X_i^4 < \infty$. Moreover, one can see that for $X_i^* = (X_i - \mu_i)$, where $\mu_i = \mathbb{E}X_i$ the proof is still correct. From Fact 4 we have

$$d_W\left(\frac{X}{\sigma}, Z\right) \leqslant \frac{D^2}{\sigma^3}\sum_{i=1}^{n}\mathbb{E}|X_i^*|^3 + \frac{D^{\frac{3}{2}}\sqrt{26}}{\sigma^2\sqrt{\pi}}\sqrt{\sum_{i=1}^{n}\mathbb{E}\left(X_i^*\right)^4}.$$

Summing it up we obtain

$$2d_K\left(\frac{X}{\sigma}, Z\right)(1+e^{\varepsilon}) + \delta_1 \leqslant 2(1+e^{\varepsilon})\left(\frac{2}{\pi}\right)^{\frac{1}{4}}\sqrt{d_W\left(\frac{X}{\sigma}, Z\right)} + \frac{5}{4\sqrt{n}} \leqslant$$

$$\leqslant c(\varepsilon)\sqrt{\frac{D^2}{\sigma^3}\sum_{i=1}^{n}\mathbb{E}|X_i^*|^3 + \frac{D^{\frac{3}{2}}\sqrt{26}}{\sigma^2\sqrt{\pi}}\sqrt{\sum_{i=1}^{n}\mathbb{E}\left(X_i^*\right)^4}} + \frac{5}{4\sqrt{n}} = \delta,$$

where

$$c(\varepsilon) = 2(1+e^{\varepsilon})\left(\frac{2}{\pi}\right)^{\frac{1}{4}}.$$

Therefore we have

$$\mathbb{P}(X \in B_u) \leqslant e^{\varepsilon}\mathbb{P}(X \in B_v) + \delta,$$

which concludes that $M(X)$ is (ε, δ)-NP. □

4.5 Adversary with Auxiliary Information

So far we have not discussed the auxiliary information of the Adversary, as we assumed that he only knows the correct distribution of the data vector and dependencies in the data (if they exist). We would like to extend our results from sections 4.3 and 4.4 to take into account the adversary's knowledge about the exact values of at most fraction γ of users. Let us assume that the auxiliary information of the Adversary consists of all records (values) of a subset Γ of the data. Let $|\Gamma| = \gamma \cdot n$. Instead of n users contributing to adversarial uncertainty, we will have $(1-\gamma)\cdot n$ users who, due to randomness in their data, make the aggregated value private. This is stated in the following observation

Observation 1. Let us consider the Adversary with knowledge of exact values of all records of a subset Γ of the data. Let $|\Gamma| = \gamma \cdot n$. All previous theorems in

this chapter can be easily adapted to such Adversary by considering data of size $(1-\gamma)n$ instead of n. This essentially captures the fact that all other users (about whom the Adversary has no information) still contribute to the adversarial uncertainty. Moreover, if we assume that the Adversary has auxiliary information about every record of the data (that is $|\Gamma| = n$) then this model collapses to standard differential privacy, where no uncertainty comes from the data itself. This shows that indeed the standard differential privacy is a special, most pessimistic, case of this model.

Let us first introduce an extension to Theorem 12, which takes into account the Adversary's knowledge about the exact values of fraction of users.

Theorem 14. *Let $X = (X_1, \ldots, X_n)$, where X_i's are independent random variables. Assume that the Adversary knows the exact values of users with indexes $\Gamma \subset [n]$, where $|\Gamma| = \gamma n$. Let $\mu_i = \mathbb{E}X_i$, $\sigma_\Gamma^2 = \frac{\sum_{i\in[n]\setminus\Gamma} \mathbb{V}ar(X_i)}{(1-\gamma)n}$ and $\mathbb{E}|X_i|^3 < \infty$ for every $i \in [n]$. Let $M(X) = \sum_{i=1}^n X_i$. Assume that $\sqrt{\frac{\Delta^2 \ln((1-\gamma)n)}{(1-\gamma)n\sigma_\Gamma^2}} < 1$, where Δ is the data sensitivity of $M(X)$. Then for ε such that*

$$\sqrt{\frac{\Delta^2 \ln((1-\gamma)n)}{(1-\gamma)n\sigma_\Gamma^2}} < \varepsilon < 1,$$

and

$$\delta = \frac{1.12 \sum_{i\in[n]\setminus\Gamma} \mathbb{E}|X_i - \mu_i|^3}{\left(\sum_{i\in[n]\setminus\Gamma} \mathbb{V}ar(X_i)\right)^{\frac{3}{2}}}(1+e^\varepsilon) + \frac{5}{4\sqrt{n}},$$

$M(X)$ is (ε, δ)-NP.

Proof Proof of this theorem is analogous to the proof of Theorem 12, with the single difference that only non-compromised users contribute to the adversarial uncertainty, namely variance of the sum consists of the uncompromised users variance. Therefore when using Berry-Esseen theorem we have smaller variance than in the case where $\gamma = 0$. □

Note that in the proof we assume that we know which subset of users is compromised. This might obviously be unknown to the data owner, so we can assume the worst case, namely that the compromised subset Γ is the subset of size γn with the greatest variance. Then the theorem holds, no matter which users are really compromised. Similarly we can introduce an extension to Theorem 13.

Theorem 15. *Let $X = (X_1, \ldots, X_n)$, where X_i's are, possibly dependent, random variables and $M(X) = \sum_{i=1}^{n} X_i$. Assume that the Adversary knows the exact values of users with indexes $\Gamma \subset [n]$, where $|\Gamma| = \gamma n$. Let $\mathbb{E}X_i = \mu_i$ and $\mathbb{E}X_i^4 < \infty$. Assume there are dependency neighborhoods N_i, $i \in [n]$, where $D = \max_{1 \leqslant i \leqslant n} |N_i|$. Let $\sigma_\Gamma^2 = \mathbb{V}ar(X \setminus \Gamma)$ and data sensitivity of $M(X)$ be Δ. Assume that $\sqrt{\frac{\Delta^2 \ln((1-\gamma)n)}{\sigma_\Gamma^2}} < 1$. Then for ε such that*

$$\sqrt{\frac{\Delta^2 \ln((1-\gamma)n)}{\sigma_\Gamma^2}} < \varepsilon < 1,$$

and

$$\delta = c(\varepsilon)\sqrt{\frac{D^2}{\sigma_\Gamma^3} M_X^3 + \frac{D^{\frac{3}{2}}\sqrt{26}}{\sigma^2\sqrt{\pi}}\sqrt{M_X^4}} + \frac{5}{4\sqrt{(1-\gamma)n}},$$

where

$$M_X^3 = \sum_{i \in [n] \setminus \Gamma} \mathbb{E}|X_i - \mu_i|^3,$$

$$M_X^4 = \sum_{i \in [n] \setminus \Gamma} \mathbb{E}\,(X_i - \mu_i)^4,$$

$$c(\varepsilon) = 2(1 + e^{\varepsilon})\left(\frac{2}{\pi}\right)^{\frac{1}{4}},$$

$M(X)$ is (ε, δ)-NP.

Proof Here also the proof is analogous to the proof of Theorem 13, and also the difference is that only non-compromised users contribute to adversarial uncertainty, namely variance of the sum consists of the uncompromised users variance. When we bound the Kolmogorov distance (using Stein's method) between the sum and a normal distribution, we use one with smaller variance (namely variance of $X \setminus \Gamma$) than in the case where $\gamma = 0$. □

As in the previous theorem, a practitioner can assume the worst case, namely that the compromised subset Γ is the subset of size γn with the greatest variance. These simple extensions of our previous theorems give us a complete insight into noiseless privacy in adversarial model presented in Subsection 4.1.2. The owner of the data (or any party responsible for the privacy in central or distributed database) can give his users a rigorously proved guarantee that as long as at most a fraction γ

of users is compromised and (in dependent case) if the size of the greatest dependent subset is at most D, then the privacy parameters are at least as good (we have shown the upper bound for the parameters) as given in Theorem 14 if the data is independent or in Theorem 15 if there are dependencies (known to the Adversary) in the data.

4.6 Synergy Between Adversarial Uncertainty and Noise Addition

In previous sections we have shown what are the privacy parameters for the randomness inherently present in the data. However, it is easy to imagine that in many cases the amount of randomness (adversarial uncertainty) might be too small to ensure desired size of privacy parameters. Does it mean that in such case we have to step back and use only standard differential privacy methods? Fortunately, it does not. It turns out that the proofs of our theorems are constructed in such a way, that it is possible to extend them to the case where we add some noise to increase the randomness in the data. Even more importantly, it is also easy to quantify how much noise has to be added to improve privacy of the data to the desired parameter in our adversarial model.

To the best of authors knowledge, so far there has not been any approach in the privacy literature to combine the idea of utilizing adversarial uncertainty (randomness in data) and standard approach which is adding appropriately calibrated noise. The idea of adding noise to already somewhat random data is quite simple, yet it needs to be carefully analysed so that one may know exactly how much does it enhance the privacy. It is intuitively very natural to think that the more randomness is present in the data, the less noise (or none, if the randomness itself is enough) we have to add to satisfy desired level of privacy. However, to become a state-of-the-art approach to preserving privacy, this intuition has to be formally introduced, rigorously quantified and proved. We introduce the following

Theorem 16. *Let $X = (X_1, \ldots, X_n)$ where X_i's are random variables such that $\mathbb{V}ar(X) = \sigma^2$. Let $M(X)$ be (ε_1, δ)-NP with data sensitivity Δ. Let ξ be an unbiased noise of variance σ_ξ^2. Then $M^*(X) = M(X + \xi)$ is (ε, δ)-NP, where*

$$\varepsilon = \sqrt{\frac{\Delta^2 \ln(n)}{\sigma^2 + \sigma_\xi^2}},$$

as long as $0 < \varepsilon < 1$.

Proof This formula can be obtained in a straightforward manner from our previous proofs. Similarly as in theorems 14 and 15 one can easily see that the sum of data with added noise has variance $\sigma^2 + \sigma_\xi^2$, because the noise is independent of the data. Therefore appropriate normal random variables to which we bound the distance of our sum (as in Berry-Esseen theorem and Stein's method) will have greater variance, which in turn gives smaller ε. □

This approach is similar to the case where the adversary has information about exact values of some fraction of the data, but this time we add variance instead of subtracting it. Improving δ parameter by adding noise seems to be more difficult, as it might require different approach to previous theorems. We leave it as an interesting problem for future work. After this theorem we can also present a useful observation

Observation 2. We can state Theorem 16 in a different way, namely for a fixed privacy parameter ε, the necessary variance of the noise to achieve desired level of privacy is

$$\sigma_\xi^2 = \max\left(\frac{\Delta^2 \ln(n) - \varepsilon^2\sigma^2}{\varepsilon^2}, 0\right).$$

This observation is obtained by using Theorem 16 and straightforward algebraic manipulations.

We also give more specific observation concerning noise having Laplace distribution, which is a common technique in standard differential privacy approach (see for example [28]).

Observation 3. Let $X = (X_1, \ldots, X_n)$ be a data vector, the data sensitivity is Δ and $\mathbb{V}ar(\sum_{i=1}^n X_i) = \sigma^2$. We consider mechanism $M(X)$ which, due to adversarial uncertainty has certain privacy parameters (ε_1, δ). We show that $M^*(X) = M(X + \xi)$ where $\xi \sim Lap(\frac{\Delta}{\varepsilon_2})$ preserves privacy with parameters (ε, δ), where

$$\varepsilon = \sqrt{\frac{\varepsilon_1^2 \cdot \varepsilon_2^2 \cdot \ln(n)}{2\varepsilon_1^2 + \varepsilon_2^2 \ln(n)}}.$$

This observation is obtained by application of Theorem 16 for $\xi \sim Lap(\frac{\Delta}{\varepsilon_2})$.

Theorem 16 allows the party responsible for preserving privacy to enhance parameter ε of the data itself by using standard methods of differential privacy. See that the noise necessary to achieve the desired level of privacy is smaller compared to using standard differential privacy methods. It is due to the fact, that we already have some level of privacy due to the inherent randomness present

in the data. We conclude our discussion concerning synergy between adversarial uncertainty and differential privacy approach by showing a following

Example 6. We consider a data vector $X = (X_1, \ldots, X_n)$ and mechanism $M(X)$ having the data sensitivity $\Delta = 10$ and $\mathbb{V}ar(M(X)) = \sigma^2 = \frac{n}{10}$. We enhance the privacy by adding Laplace noise of variance σ_ξ^2. Using Theorem 16 and Observation 2 we can compute what is the necessary variance of noise to obtain privacy parameter $\varepsilon = 0.2$ depending on the number of users. See Figure 4.5. See that we have also plotted the variance of noise using differential privacy approach, namely Laplace mechanism (see [28]). We can see that in this example, for n up to around 1050 we have to apply standard differential privacy mechanism. Moreover, for n greater than approximately 1350 we know from our previous results that noise is unnecessary, because the data has sufficient privacy parameters due to inherent randomness. Most interesting, in terms of synergy of adversarial uncertainty and differential privacy methods is the case where n is between 1050 and 1350. Here one can see that adding significantly less noise than using standard differential privacy approach is sufficient to obtain desired parameter $\varepsilon = 0.2$.

To sum up all our results, we present a flowchart, which shows on high level of abstraction how should the data owner approach the problem of preserving privacy in a general manner. See Figure 4.6.

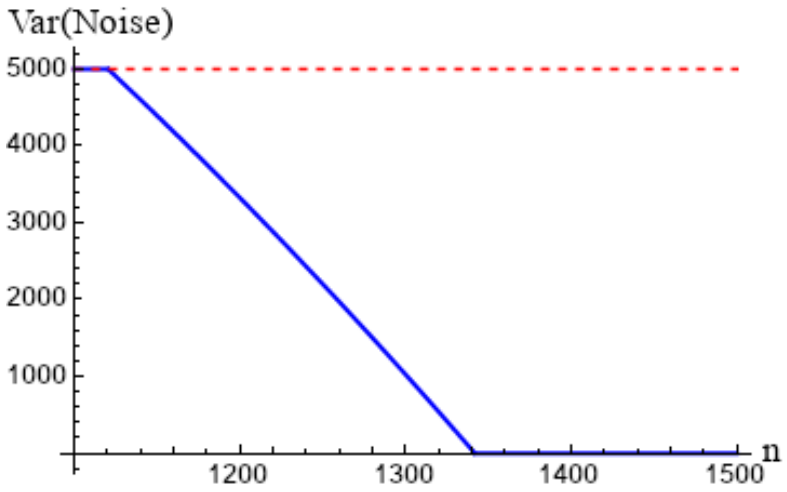


Figure 4.5: Example 6, red dashed line shows the variance of necessary noise for Laplace mechanism using standard differential privacy. Blue thick line shows the variance of necessary noise after taking into account the adversarial uncertainty.

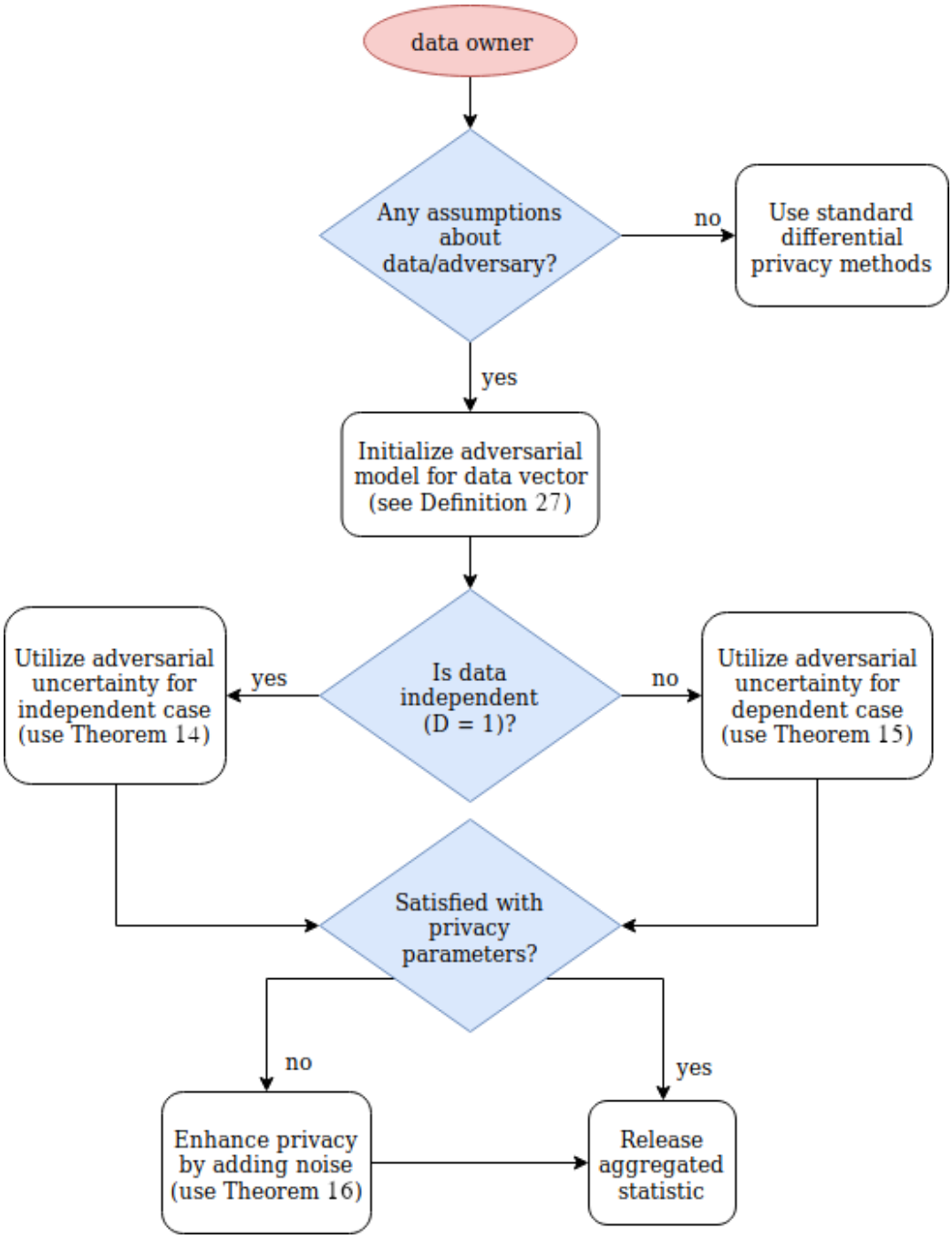


Figure 4.6: A flowchart for privacy preserving in a general way.

4.7 Applications

- The notion of noiseless privacy and our bounds for privacy parameters are useful especially in distributed case for two reasons. First, in distributed systems quite often the noises which have to be added by users render the data practically useless (too much disturbance). Second, in such systems it is more common to assume that the Adversary does not have full knowledge, i.e. can know only some fraction of the data. See that, if the noiseless privacy assumptions are met and the privacy parameters are satisfying, one could for example run protocols from [62, 17] with only the cryptographic part, without adding noises to the values. The noises added in standard approach turn out to be too large for practical applications in various scenarios (see [39]).

- The idea of noiseless privacy can be used for a wide range of applications including networks of sensing environmental parameters, smart metering (e.g, electricity) or clinical research. Most important, however, is that in all these areas there are natural cases, where we can make some assumptions about the knowledge of the Adversary.

- Assume we have a cloud service which holds shopping preferences of its users. The data is distributed amongst many servers which are completely separated from each other. We assume that some of these servers became compromised, which means that, say at most 50 percent of the values are known to the Adversary. We might know that the greatest dependent subset of our data has size at most D. This yields model ($Adv_X(D, \gamma)$) for known (or at least upper bounded) γ, D and distributions of the rest of the data.

Chapter 5

Summary

In this thesis we focused on improving privacy in unreliable distributed systems under differential privacy regime. We achieved it by proposing new protocols, both for privacy preserving aggregation and for improving the density of graph, which in turn improves privacy. We also proposed a different approach to make privacy more practical, which can decrease the magnitude of noise needed for data aggregation. This might be especially useful in unreliable distributed systems, as they tend to require a lot of noise to maintain satisfying privacy parameters.

- Our fault tolerant, privacy preserving aggregation protocol offers much better precision than current solutions. In order to obtain this, we allowed limited communication between the nodes. Moreover, the protocol greatly benefits from having sufficiently dense network of users. This assumption deviates from the classic model. We provided a precise analysis of accuracy of the data aggregation protocol presented in [17]. It turned out that, in many cases, its accuracy may not be sufficient even if the number of faults is moderate. We experimentally compared both protocols. From the theoretical point of view the important question is about the possible trade-offs between privacy protection, volume of communication and possible accuracy of the results of aggregation.

- Above mentioned protocol works best if performed on a dense graph. We presented how to improve the size of the largest connected component under massive adversarial attack and demonstrated why this observation is important for a wide range of applications. Moreover, our methods are conceptually simple and can be performed **locally**, i.e. with minimal knowledge

about the global network. We proved that the presented methods are efficient in preferential-attachment graphs, which are commonly believed to be an accurate model of various real-life networks including social interaction networks, World Wide Web, airline networks and many other. Finally, we confirmed our observations using experiments on graphs of real networks. Note also that our protocols improve security of participating individuals, but the level of privacy is improved also for other users.

- We have also taken a slightly different approach at making privacy more practical. Namely, we continued already existing work concerning relaxation of differential privacy called noiseless privacy, which utilizes adversarial uncertainty. We have shown an explicit bounds for privacy parameters. We have presented specific model of privacy (similar, but more practically useful than the one given in the seminal paper [11]) and introduced model of the adversary. To the best of our knowledge, in the papers concerning leveraging inherent randomness in the data there were only asymptotic results so far. By showing an **explicit bounds** for privacy parameters, we have made the whole idea more approachable in practice. Another important contribution of this paper is approaching **dependent** data, namely using the notion of dependency neighborhoods. We give privacy parameters bounds for **any** distribution and a wide class of dependencies. The data owner only has to plug the variance of the data (or the lower bound for variance), data sensitivity (which is also necessary in standard differential privacy approach) and appropriate central moments. Then he can give a specific privacy guarantee to its users that as long as at most γ is compromised and as long as the greatest dependent subset has size D. We wanted to make these theorems usable not only by the privacy experts, but any developer or specific domain expert. Furthermore, we have shown how does the standard differential privacy approach combines with the notion of inherent randomness in the data. It turns out that if the data is more 'random', then less noise is necessary to achieve specific privacy parameter.

Our goal for all these techniques, used separately or combined, was to make them not only correct and interesting from the theoretical point of view, but also relatively easy to use in practice. Therefore, they could lead to faster adoption of mathematically rigorous approach to privacy preservation in practical applications.

Bibliography

[1] Gergely Ács and Claude Castelluccia, I have a dream! (differentially private smart metering), International Workshop on Information Hiding, Springer, 2011, pp. 118–132.

[2] Réka Albert and Albert-László Barabási, Statistical mechanics of complex networks, Reviews of Modern Physics **74** (2002), no. 1, pp. 47–101.

[3] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg, Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography, Proceedings of the 16th ACM International Conference on World Wide Web, 2007, pp. 181–190.

[4] Borja Balle, James Bell, Adria Gascon, and Kobbi Nissim, The privacy blanket of the shuffle model, arXiv preprint arXiv:1903.02837 (2019).

[5] Albert-László Barabási, Scale-free networks: a decade and beyond, Science **325** (2009), no. 5939, pp. 412–413.

[6] Albert-László Barabási and Eric Bonabeau, Scale-free networks, Scientific American **288** (2003), no. 5, pp. 50–59.

[7] Andrew Barbour and Louis Hsiao Yun Chen, An introduction to Stein's method, vol. 4, World Scientific, 2005.

[8] Raef Bassily, Adam Groce, Jonathan Katz, and Adam Smith, Coupled-worlds privacy: Exploiting adversarial uncertainty in statistical data privacy, IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS), 2013, pp. 439–448.

[9] Raef Bassily and Adam Smith, Local, private, efficient protocols for succinct histograms, Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC), 2015, pp. 127–135.

[10] Alina Beygelzimer, Geoffrey Grinstein, Ralph Linsker, and Irina Rish, Improving network robustness by edge modification, Physica A: Statistical Mechanics and its Applications **357** (2005), no. 3, pp. 593–612.

[11] Raghav Bhaskar, Abhishek Bhowmick, Vipul Goyal, Srivatsan Laxman, and Abhradeep Thakurta, Noiseless database privacy, International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2011, pp. 215–232.

[12] Béla Bollobás, Random graphs, Modern Graph Theory, Springer, 1998, pp. 215–252.

[13] Béla Bollobás and Oliver Riordan, The diameter of a scale-free random graph, Combinatorica **24** (2004), no. 1, pp. 5–34.

[14] Béla Bollobás, Oliver Riordan, Joel Spencer, Gábor Tusnády, et al., The degree sequence of a scale-free random graph process, Random Structures & Algorithms **18** (2001), no. 3, pp. 279–290.

[15] Béla Bollobás and Oliver M. Riordan, Mathematical results on scale-free random graphs, Handbook of Graphs and Networks: From the Genome to the Internet (2003), pp. 1–34.

[16] Joseph Calandrino, Ann Kilzer, Arvind Narayanan, Edward Felten, and Vitaly Shmatikov, "You might also like": Privacy risks of collaborative filtering, 2011 IEEE Symposium on Security and Privacy, 2011, pp. 231–246.

[17] T.-H. Hubert Chan, Elaine Shi, and Dawn Song, Privacy-preserving stream aggregation with fault tolerance., Financial Cryptography, Lecture Notes in Computer Science, vol. 7397, Springer, 2012, pp. 200–214.

[18] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev, Distributed differential privacy via shuffling, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2019, pp. 375–403.

[19] Aaron Clauset, Cosma Rohilla Shalizi, and Mark Newman, Power-law distributions in empirical data, SIAM review **51** (2009), no. 4, pp. 661–703.

[20] Damien Desfontaines, Andreas Lochbihler, and David Basin, Cardinality estimators do not preserve privacy, Proceedings on Privacy Enhancing Technologies **2019** (2019), no. 2, pp. 26–46.

[21] Irit Dinur and Kobbi Nissim, Revealing information while preserving privacy, Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 2003, pp. 202–210.

[22] Devdatt P Dubhashi and Alessandro Panconesi, Concentration of measure for the analysis of randomized algorithms, Cambridge University Press, 2009.

[23] John Duchi, Michael Jordan, and Martin Wainwright, Local privacy and statistical minimax rates, IEEE 54th Annual Symposium on Foundations of Computer Science, 2013, pp. 429–438.

[24] John Duchi, Martin Wainwright, and Michael Jordan, Local privacy and minimax bounds: Sharp rates for probability estimation, Advances in Neural Information Processing Systems, 2013, pp. 1529–1537.

[25] Cynthia Dwork, Differential privacy: A survey of results, Theory and Applications of Models of Computation, Springer, 2008, pp. 1–19.

[26] Cynthia Dwork, Differential privacy, Encyclopedia of Cryptography and Security (2011), pp. 338–340.

[27] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith, Calibrating noise to sensitivity in private data analysis, Proceedings of 3rd Theory of Cryptography Conference (TCC), 2006, pp. 265–284.

[28] Cynthia Dwork and Aaron Roth, The algorithmic foundations of differential privacy, Foundations and Trends in Theoretical Computer Science **9** (2014), no. 3-4, pp. 211–407.

[29] Keita Emura, Hayato Kimura, Toshihiro Ohigashi, and Tatsuya Suzuki, Privacy-preserving aggregation of time-series data with public verifiability from simple assumptions and its implementations, The Computer Journal **62** (2018), no. 4, pp. 614–630.

[30] Zekeriya Erkin, Juan Ramón Troncoso-Pastoriza, Reginald Lagendijk, and Fernando Pérez-González, Privacy-preserving data aggregation in smart

metering systems: An overview, IEEE Signal Processing Magazine **30** (2013), no. 2, pp. 75–86.

[31] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta, Amplification by shuffling: From local to central differential privacy via anonymity, Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, 2019, pp. 2468–2479.

[32] Abraham Flaxman, Alan Frieze, and Juan Vera, Adversarial deletion in a scale free random graph process, Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, 2005, pp. 287–292.

[33] Babak Fotouhi and Michael G. Rabbat, Degree correlation in scale-free graphs, European Physical Journal B **86** (2013), pp. 510–530.

[34] Philippe Golle, Markus Jakobsson, Ari Juels, and Paul Syverson, Universal re-encryption for mixnets, Cryptographers‘ Track at the RSA Conference, Springer, 2004, pp. 163–178.

[35] Krzysztof Grining and Marek Klonowski, Towards extending noiseless privacy: Dependent data and more practical approach, Proceedings of the 12th ACM on Asia Conference on Computer and Communications Security (AsiaCCS), 2017, pp. 546–560.

[36] Krzysztof Grining, Marek Klonowski, and Małgorzata Sulkowska, How to cooperate locally to improve global privacy in social networks? On amplification of privacy preserving data aggregation, Proceedings of the 16th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2017, pp. 464–471.

[37] Krzysztof Grining, Marek Klonowski, and Małgorzata Sulkowska, Stronger trust and privacy in social networks via local cooperation, Journal of Complex Networks (2019).

[38] Krzysztof Grining, Marek Klonowski, and Piotr Syga, Practical fault-tolerant data aggregation, International Conference on Applied Cryptography and Network Security (ACNS), Springer, 2016, pp. 386–404.

[39] Krzysztof Grining, Marek Klonowski, and Piotr Syga, On practical privacy preserving fault tolerant data aggregation, International Journal of Information Security **18** (2019), no. 3, pp. 285–304.

[40] Justin Hsu, Sanjeev Khanna, and Aaron Roth, Distributed private heavy hitters, International Colloquium on Automata, Languages, and Programming, Springer, 2012, pp. 461–472.

[41] Matthew Joseph, Aaron Roth, Jonathan Ullman, and Bo Waggoner, Local differential privacy for evolving data, Advances in Neural Information Processing Systems, 2018, pp. 2375–2384.

[42] Shiva Prasad Kasiviswanathan, Homin Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith, What can we learn privately?, SIAM Journal on Computing **40** (2011), no. 3, pp. 793–826.

[43] Daniel Kifer and Ashwin Machanavajjhala, Pufferfish: A framework for mathematical privacy definitions, ACM Transactions on Database Systems (TODS) **39** (2014), no. 1, pp. 1–36.

[44] Jure Leskovec and Andrej Krevl, SNAP Datasets: Stanford large network dataset collection, `http://snap.stanford.edu/data`, June 2014.

[45] Jure Leskovec, Kevin Lang, Anirban Dasgupta, and Michael Mahoney, Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters, Internet Mathematics **6** (2009), no. 1, pp. 29–123.

[46] Ninghui Li, Wahbeh Qardaji, and Dong Su, On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy, Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, 2012, pp. 32–33.

[47] Katrina Ligett, Seth Neel, Aaron Roth, Bo Waggoner, and Steven Wu, Accuracy first: Selecting a differential privacy level for accuracy constrained ERM, Advances in Neural Information Processing Systems, 2017, pp. 2566–2576.

[48] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam, ℓ-diversity: Privacy beyond κ-anonymity, Proceedings of the 22nd International Conference on Data Engineering, 2006, pp. 24–36.

[49] Julian J McAuley and Jure Leskovec, Learning to discover social circles in ego networks., NIPS, vol. 2012, 2012, pp. 548–556.

[50] Michael Mitzenmacher and Eli Upfal, Probability and computing: randomization and probabilistic techniques in algorithms and data analysis, Cambridge University Press, 2017.

[51] Arvind Narayanan, Elaine Shi, and Benjamin Rubinstein, Link prediction by de-anonymization: How we won the kaggle social network challenge, International Joint Conference on Neural Networks, 2011, pp. 1825–1834.

[52] Arvind Narayanan and Vitaly Shmatikov, Robust de-anonymization of large sparse datasets, 29th IEEE Symposium on Security and Privacy, 2008, pp. 111–125.

[53] Arvind Narayanan and Vitaly Shmatikov, De-anonymizing social networks, arXiv preprint arXiv:0903.3276 (2009).

[54] Arvind Narayanan and Vitaly Shmatikov, Myths and fallacies of personally identifiable information, Communications of the ACM **53** (2010), no. 6, pp. 24–26.

[55] Andreas Pfitzmann and Marit Köhntopp, Anonymity, unobservability, and pseudonymity - a proposal for terminology, Designing Privacy Enhancing Technologies, Springer, 2001, pp. 1–9.

[56] Iosif Pinelis, Characteristic function of the positive part of a random variable and related results, with applications, Statistics & Probability Letters **106** (2015), pp. 281–286.

[57] Vibhor Rastogi and Suman Nath, Differentially private aggregation of distributed time-series with transformation and encryption, Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, pp. 735–746.

[58] Matthew Richardson, Rakesh Agrawal, and Pedro Domingos, Trust management for the semantic web, International Semantic Web Conference, Springer, 2003, pp. 351–368.

[59] Nathan Ross, Fundamentals of Stein's method, Probab. Surv **8** (2011), pp. 210–293.

[60] Pierangela Samarati, Protecting respondents identities in microdata release, IEEE Transactions on Knowledge and Data Engineering **13** (2001), no. 6, pp. 1010–1027.

[61] Pierangela Samarati and Latanya Sweeney, Generalizing data to provide anonymity when disclosing information, Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), 1998, p. 188.

[62] Elaine Shi, Richard Chow, T-H. Hubert Chan, Dawn Song, and Eleanor Rieffel, Privacy-preserving aggregation of time-series data, Proceedings of the 18th Network and Distributed System Security Symposium (NDSS), 2011.

[63] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov, Membership inference attacks against machine learning models, 38th IEEE Symposium on Security and Privacy, 2017, pp. 3–18.

[64] Steven Strogatz, Exploring complex networks, Nature **410** (2001), no. 6825, pp. 268–276.

[65] Latanya Sweeney, Weaving technology and policy together to maintain confidentiality, The Journal of Law, Medicine & Ethics **25** (1997), no. 2-3, pp. 98–110.

[66] Latanya Sweeney, k-anonymity: a model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10** (2002), pp. 557–570.

[67] I.S. Tyurin, A refinement of the remainder in the lyapunov theorem, Theory of Probability & Its Applications **56** (2012), no. 4, pp. 693–696.

[68] Remco van der Hofstad, Random graphs and complex networks, Cambridge Series in Statistical and Probabilistic Mathematics (2016).

[69] WolframResearch, Hypergeometric2F1, 2011, `http://functions.wolfram.com/HypergeometricFunctions/Hypergeometric2F1`.

[70] Xiaokui Xiao and Yufei Tao, M-invariance: towards privacy preserving re-publication of dynamic datasets, Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, pp. 689–700.

[71] Yang Yang, Zhoujun Li, Yan Chen, Xiaoming Zhang, and Senzhang Wang, Improving the robustness of complex networks with preserving community structure, PLOS ONE **10** (2015), no. 2.

[72] Haotian Zhang, Elaheh Fata, and Shreyas Sundaram, A notion of robustness in complex networks, IEEE Transactions on Control of Network Systems **2** (2015), no. 3, pp. 310–320.

[73] Jichang Zhao and Ke Xu, Enhancing the robustness of scale-free networks, Journal of Physics A: Mathematical and Theoretical **42** (2009), no. 19.