Discrete Mathematics and Theoretical Computer Science ....
.... **vol. 28:2 #14 (2026)**

Probabilistic Counters for Privacy Preserving Data Aggregation[*]

Dominik Bojko[1] Krzysztof Grining Marek Klonowski[2]

[1] *Wrocław University of Science and Technology, Faculty of Fundamental Problems of Technology, Department of Fundamentals of Computer Science, Wrocław, Poland*

[2] *Wrołcaw University of Science and Technology, Faculty of Fundamental Problems of Technology, Department of Artificial Intelligence, Wrocław, Poland*

Probabilistic counters are well-known tools often used for space-efficient set cardinality estimation. In this paper, we investigate probabilistic counters from the perspective of preserving privacy. We use the standard, rigid differential privacy notion. The intuition is that the probabilistic counters do not reveal too much information about individuals but provide only general information about the population. Therefore, they can be used safely without violating the privacy of individuals. However, it turned out, that providing a precise, formal analysis of the privacy parameters of probabilistic counters is surprisingly difficult and needs advanced techniques and a very careful approach.

We demonstrate that probabilistic counters can be used as a privacy protection mechanism without extra randomisation. That is, the inherent randomisation of the protocol is sufficient to protect privacy, even if the probabilistic counter is used multiple times. In particular, we present a specific privacy-preserving data aggregation protocol based on Morris Counter and MaxGeo Counter. Some of the results presented are devoted to counters that have not been investigated so far from the perspective of privacy protection. Another part is an improvement of the previous results. We show how our results can be used to perform distributed surveys and compare the properties of counter-based solutions and a standard Laplace method.

Keywords: probabilistic counter, Morris counter, differential privacy

1 Introduction

Since Big Data related topics have been widely developed in recent years, solutions that focus on saving memory resources have become very popular. We would like to consider a standard example of such space-efficient mechanisms, namely probabilistic counters, which are used to represent the cardinality of dynamically counted events. More precisely, we would like to indicate the occurrence of n events using a very small (significantly less than $\log n$) number of bits. We assume that n is unknown in advance and may change. Clearly, a simple information-theoretic argument convinces us that it is not feasible if we demand an exact representation of the number of events. Nevertheless, there are some very efficient solutions that require only $\Theta(\log\log n)$ bits and guarantee sufficient accuracy for a wide range of applications. As

[*] Supported by Polish National Science Center grant number UMO-2018/29/B/ST6/02969UMO-2018/29/B/ST6/02969

ISSN 1365–8050

examples, one can point most *probabilistic counters* – probabilistic structures well known in the literature since the seminal Morris' paper Morris (1978) followed by its thorough mathematical analysis by Flajolet in Flajolet (1985). They are used as building blocks in many space-efficient algorithms in the field of data mining or distributed data aggregation in networks or smart metering, just to mention a few (Baquero et al. (2009) or Cichoń and Gotfryd (2018)).

In this paper, we investigate probabilistic counters from the privacy-protection perspective. Our analysis is based on a differential privacy notion, which is commonly considered the only state-of-the-art approach. The differential privacy has the undeniable advantage of being mathematically rigorous and formally provable, contrary to previous anonymity-derived privacy definitions. This approach to privacy-preserving protocols can be used to give a formal guarantee for privacy that is resilient to any form of post-processing. For a survey about differential privacy properties, see Dwork and Roth (2014) and references therein. Analysis of protocols based on differential privacy is usually technically complex, but by using this notion, we are immune to, e.g., linkage attacks (see, for example Narayanan and Shmatikov (2009, 2010)).

The idea behind differential privacy is as follows: for two "neighbouring" scenarios that differ only in the participation of a single user, a differentially private mechanism should provide a response chosen from very similar distributions. Roughly speaking, differential privacy is described by two parameters: ε – which controls a similarity of probabilities of common events – and δ – which is related to a probability of unusual events. The smaller the parameters, the better from the privacy point of view. In effect, judging by the output of the mechanism, one cannot say if a given individual (user) was taken into account for producing a given output. Intuitively, probabilistic counters should provide a high level of differential privacy since, statistically, many various numbers of events are "squeezed" into a small space of possible output results. In the case of one counter considered in our paper (MaxGeo) counter, one can find some similar, recent results about the privacy the algorithm offers. Nevertheless, the question about the value of the parameters of the potential differential privacy property remains open (see the discussion in Section 2). Moreover, when considering a small number of events n, an additional problem may occur, as it may be possible to distinguish that the number of events is different from $n-1$ or $n+1$ with a significant probability.

In our paper, we provide a very precise analysis of two well known probabilistic counters from the perspective of preserving privacy. It turned out that this task is surprisingly complex from the mathematical point of view. Our primary motivation is to find possibly accurate privacy parameters for the two most fundamental probabilistic counter protocols, namely the Morris Counter Morris (1978) and the MaxGeo Counter Szpankowski and Rego (1990). Note that the second one is used for yet another popular algorithm — HyperLogLog Flajolet et al. (2007). One may realise that these two counters are relatively old; however, they, together with their modifications, have been extensively used until these days. Morris Counter is often used in big data solutions, for instance, to measure network's capabilities Einziger et al. (2018). The most crucial examples of refinements of the HyperLogLog algorithm are mentioned in Section 2.

We claim that a high-precision analysis in the case of probabilistic counters is particularly important. This is because even a mechanism with very good privacy parameters can cause a serious privacy breach when used multiple times. That is, privacy loss/information leakage accumulates over multiple releases (see, e.g., Dwork and Roth (2014)). Probabilistic counters in realistic scenarios may be used as fundamental primitives and subroutines in more complex protocols, since the differential privacy property is immune to post-processing.

We also show that those two probabilistic counters can be used safely without any additional randomisation, even in very demanding settings. It is commonly known that no deterministic algorithm can provide non-trivial differential privacy. However, Probabilistic counters have inherent randomness, achieving the desired privacy parameters. In other words, one can say that probabilistic counters are safe by design, and we do not need any additional privacy-orientated methods. In particular, what is most important, existing, working implementations do not need to be changed if we start demanding the provable privacy of a system.

Finally, we demonstrate how our results can be used for constructing a data aggregation protocol based on probabilistic counters that can be used in some specific scenarios until we want them to satisfy even more rigorous privacy properties.

To the best of our knowledge, most of the results are new and deal with protocols not considered before in the context of privacy preservation. Some other (such as the MaxGeo counter) improve some previous results (e.g. Smith et al. (2020)).

1.1 Paper structure and results

Starting from this point, for the sake of clarity, we use the abbreviation *DP* as a shortcut for differential privacy, while this property is described by some parameters.

The main contribution of our paper is as follows:

- We prove that the classic Morris Counter satisfies $(\varepsilon(n), \delta(n))$-DP with $\varepsilon(n) = O\left(\frac{(\log(n))^2}{n}\right)$ and $\delta(n) = O\left(\max\left\{n^{-(\ln(n))^{c-1}}, n^{-1}\left(\ln(n)\right)^{-c}\right\}\right)$, for any $c > 0$ (Theorem 2 in Section 6).
- We prove that the Morris Counter satisfies the $(L(n), 0.00033)$-DP property (see Definition 2), where $L(n) = -\ln\left(1 - 16/n\right) \approx 16/n$ (Theorem 1 in Section 6). In Observation 2, we also show that the constant 16 cannot be improved.
- We prove that MaxGeo Counter satisfies the property (ε, δ) -DP (Definition 2 is provided in Section 3) if the number of events n (in Section 4 the concept of event is clarified) is at least $\dfrac{\ln(\delta)}{\ln\left(1 - 2^{-l_\varepsilon}\right)}$, where $l_\varepsilon = \left\lceil \log\left(\frac{e^\varepsilon}{e^\varepsilon - 1}\right)\right\rceil$ (Theorem 3 in Section 4).
- We construct a distributed survey protocol to preserve privacy based on probabilistic counters in Section 5 and compare it with the Laplace method, which is considered as the actual state of the art of differentially private protocols and is not based on probabilistic counters.

The remainder of this paper is organised as follows. First, in Section 2 we mention work related to our paper and some popular examples of other probabilistic counters, which are not considered in this paper. In Section 3, we recall the differential definition of privacy. In Section 4 we describe probabilistic counters; further, we recall the definitions of both Morris and MaxGeo Counters. Moreover, we state Fact 1, a useful reformulation of the standard definition of differential privacy for probabilistic counters. In Section 5, we demonstrate how a probabilistic counter can be used to construct a data aggregation protocol in a very particular, yet natural scenario. Section 6 consists of formulations of our main results for both counters. Section 7 gives some ideas about the realisation of the scenario and the comparison of both counters. We also compare these solutions with the standard Laplace method (Section 7). For convenience of the reader, we provide the proofs in Section 8, as they are rather technical. For more clarity, some proofs and lemmas are moved to A. Finally, in Section 9 we present conclusions and future work propositions.

2 Previous and Related Work

In our paper, we provide a detailed analysis of some probabilistic counters from the perspective of differential privacy. Differential privacy concepts have been discussed in many papers in recent years. One can also find a well developed body of literature devoted to probabilistic counters and similar structures. Therefore, we limit the related literature review to the most relevant papers.

Differential Privacy literature In our paper, we focus on the inherent privacy guarantees of some probabilistic structures defined as differential privacy. The idea of differential privacy has been introduced for the first time in Dwork et al. (2006b); however, its precise formulation in the widely used form appeared for the first time in Dwork (2006). There is a long list of papers concerning differential privacy, e.g. Dwork et al. (2006a); Dwork and Lei (2009); Dwork et al. (2010), to mention a few.

Most of these papers focus on a centralised (global) model, namely, a database with a trusted party holding it. See that in our paper, despite the distributed setting, we have the same (non-local) trust model. In particular, we assume an existence of a *curator* that is entitled to gather and see all participants' data in the clear and release the computed data to a wider (possibly untrusted) audience. Comprehensive information on differential privacy can be found in Dwork and Roth (2014).

Probabilistic counters and their applications The idea of probabilistic counters, along with the well-known Morris Counter was presented in the seminal paper Morris (1978). The aim was to construct a very small data structure to represent a large set of events of some kind. In our paper, we focus on the Morris Counter analysed in detail in Flajolet (1985). The second structure discussed in our paper is MaxGeo Counter, introduced and analysed in Szpankowski and Rego (1990). More detailed and precise analysis can be found in Eisenberg (2008). The most important application of MaxGeo Counter can be found in Flajolet et al. (2007), where the authors propose the well-known HyperLogLog algorithm. Its practical applications are widely described in Heule et al. (2013). There are several widely used improvements of the HyperLogLog algorithm: HyperLogLog+ Heule et al. (2013), Streaming HyperLogLog with sketches based on historical inverse probability Cohen (2015) or martingal estimator Ting (2014) or empirically adjusted HyperBitBit (proposed by R. Sedgewick Sedgewick (2018)). The main goal of these adjustments is to reduce the memory requirements (see, e.g., Yun et al. (2015) or Ting (2019)). For instance, some of the above solutions are used in database systems for query' optimisation or for document classification purposes. Moreover, the MaxGeo counter was used in Palmer et al. (2002), for an adjustment of the ANF tool, developed for data mining from extensive graphs, which enables it to answer many different questions based on some neighbourhood function defined on the graph.

Unsurprisingly, one of the main applications of the approximate counter is to compute the size of a database or its specific subset. A set of such applications can be found in Flajolet and Martin (1985). In Van Durme and Lall (2009), the authors use Morris Counter for online, probabilistic, and space-efficient counting over streams of fixed, finite length. The authors of Cichon and Macyna (2011) proposed an application of a Morris Counter system for flash memory devices. Another application, presented in Csűrös (2010), is a revisit of Morris Counter designed for binary floating-point numbers. In Gronemeier and Sauerhoff (2009), Morris Counter is used in a well-known problem of counting frequency moments of long data streams. The authors of Dice et al. (2013) focused on making probabilistic counters scalable and accurate in concurrent settings. The paper on probabilistic counters in hardware can be found in Riley and Zilles (2006). A slightly modified version of Morris Counter called Morris+ was recently introduced in Nelson and Yu (2022) with the proof of its optimality in terms of accuracy–memory trade-off.

In random graphs theory, Morris Counter is usually connected to greedy structures. For instance, in an arrangement of a randomly labelled graph in Gilbert model $G(n, p)$, it is possible to construct a greedy stable set S_n, which size has the same distribution as the Morris Counter M_n of the base $a = (1 - p)^{-1}$ (see, e.g., Frieze and Karonski (2015) or Bollobás (1998) for the fundamentals of random graph theory).

There are many other birth processes that are quite similar to the Morris Counter, which are applicable in a variety of disciplines like biology, physics, or the theory of random graphs. Short descriptions of such examples can be found in Crippa and Simon (1997). When talking about probabilistic counters, it is worth mentioning the Bloom filter Bloom (1970), which are space-efficient probabilistic data structures that are representations of sets. There exists a probabilistic counter that approximates the number of elements represented by the given Bloom filter Swamidass and Baldi (2007).

Other common examples of probabilistic counters are F_p counters Alon et al. (1999); Indyk (2000), which approximate the p-th moments of frequencies of occurrences of different elements in the database. Let us also mention a paper Mishra and Sandler (2006) in which one can find numerous applications of similar constructions to create pseudorandom sketches in Big Data algorithms.

Notice that the variety of possible applications of probabilistic counters creates an opportunity to exploit inherent differential privacy properties. However, a new challenge arises — to calculate the parameters of differential privacy for those counters, which are not connected straightforwardly with Morris or MaxGeo Counters.

Probabilistic counters and preserving privacy Some probabilistic counters and similar structures were previously considered in terms of privacy preservation. We mention only the papers strictly related to the algorithms discussed in our paper (i.e., Morris Counter and MaxGeo). The authors of Desfontaines et al. (2019) show that in the scenario of using different types of probabilistic counters for set cardinality estimation with the Adversary being able to extract the intermediate values of the counter, privacy is not preserved. Note that in this paper, we perform data aggregation instead of cardinality estimation. Moreover, we assume the Adversary is not able to extract any intermediate values from the counter. That is, we consider a **global** model, while the result from Desfontaines et al. (2019) assumes the settings closer to the classic local model Dwork and Roth (2014).

One of the main results of this submission is a careful and tight analysis of Morris Counter from the context of preserving privacy. To the best of our knowledge, such analises have not been provided so far. Our second contribution is an analogous analysis of the MaxGeo. There are a few very recent papers presenting privacy-preserving protocols that use the Flajolet–Martin sketch as a building block. In Flajolet and Martin (1985), so called Flajolet—Martin sketch was introduced. In Smith et al. (2020) authors consider a general concept of a probabilistic counter, based on several MaxGeo counters and its differential privacy. Nevertheless, they incorrectly call it a generalisation of the Flajolet–Martin sketch (they probably confuse the Flajolet–Martin sketch with LogLog sketch).

We concern a concept of MaxGeo counter, which is a core of LogLog or HyperLogLog sketch, however, it can be used in other arrangements as well. These papers in some cases provide an analysis of the privacy guaranteed by Flajolet—Martin with the global model. In all the cases, the conclusion is positive in the sense that the protocol itself provides some level of differential privacy without adding extra randomness. Beneath papers provide an analysis of privacy guaranteed by the sketches related to LogLog algorithm, which can be seen as the processing of the fundamental MaxGeo counters. From this point of view, our contribution about MaxGeo counters has a larger applicative potential.

In Smith et al. (2020) the authors consider, among others, a sketch that can be seen as a particular application of the MaxGeo counter. They introduce its differentially private version via trick (adding artificial utilities) and provide its accuracy when used to count the number of elements in multisets. Accidentally, a proof of the basic theorem from Smith et al. (2020) uses an incorrect argument (inappropriate utilisation of Hoeffding's inequality), so it is difficult to compare the results precisely. Nevertheless, the overlap of results between our paper and Smith et al. (2020) is only partial.

In Choi et al. (2020), the authors consider the LogLog sketch as a subroutine. After a careful analysis, they show that it is asymptotically $(\varepsilon, \delta = \mathrm{negl}(\lambda))$-DP (with respect to the numbers of different elements), when the number of elements counted by the mechanism is at least $8K\lambda\max(\frac{1}{\varepsilon}, 1)$, where K is some accuracy parameter, λ is some security parameter and $\mathrm{negl}(x)$ is some negligible function of argument x (Theorem 4.2 in Choi et al. (2020)). Nevertheless, the analysis does not explain how to choose parameters K and λ in order to obtain (ε, δ)-DP for a given ε and δ parameters. Moreover, a consideration of asymptotic behaviour (with respect to the number of unique elements n) is not relevant when the hash function restricts the possible result to the size bounded by its domain. Our analysis of the MaxGeo counter provides an exact (non-asymptotic) dependence between n and the parameters ε and δ.

We also mention that some other pseudorandom structures have been analysed from the perspective of differential privacy. For example, in Klonowski and Piotrowska (2018), the authors considered Bloom filters as a means of constructing a privacy-preserving aggregation protocol.

3 Differential Privacy Preliminaries

In this section, we briefly recall *differential privacy*. For more details, see, e.g. Dwork and Roth (2014). We denote the set of (positive) natural numbers by $\mathbb{N}$ and the set of all integers by $\mathbb{Z}$. Moreover, let $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. For $a, b \in \mathbb{Z}$ let us define a discrete interval $[a, b] \cap \mathbb{Z}$ by $[a : b]$. We also define $[n] = \{1, 2, \ldots, n\}$ for $n \in \mathbb{N}$. We assume that there exists a trusted *curator* who holds, or securely obtains, the data of *individuals* in a (possibly distributed) database x. Every row of x consists of the data of some individual. By $\mathcal{X}$, we denote the space of all possible rows. The goal is to protect the data of every single individual, even if all users except one collude with an *adversary* to breach the privacy of this single, uncorrupted user. On the other hand, the curator is responsible for producing a *release* – a possibly accurate response to a requested *query*. This response is then released to the public, who is allowed to perform a statistical analysis on it. The differential privacy is, by design, resilient to post-processing attacks, so even if the adversary obtains the public release, he will not be able to infer anything about specific individuals participating in this release.

For simplicity, we interpret databases as their histograms in $\mathbb{N}_0^{|\mathcal{X}|}$, so we can focus only on unique rows and the numbers of their occurrences.

Definition 1 (Distance between databases) *The ℓ_1 distance between two databases $x, y \in \mathbb{N}_0^{|\mathcal{X}|}$ is defined as*

$$\|x - y\|_1 = \sum_{i \in \mathcal{X}} |x_i - y_i|,$$

where x_i and y_i denote the numbers of occurrences of an item (an individual) i in the databases x and y, respectively.

One can easily see that $\|x-y\|_1$ measures how many records differ between x and y. Moreover, $\|x\|_1$ measures the size of the database x.

A *privacy mechanism* is a randomised algorithm used by the curator that takes a database as input and produces the output (the release) using randomisation.

Definition 2 (Differential Privacy – from Dwork and Roth (2014)) *A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε,δ)-differentially private (or (ε,δ)-DP), if for all $\mathcal{S} \subseteq$ Range($\mathcal{M}$) and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x-y\|_1 \leqslant 1$ the following condition is satisfied:*

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

where the probability space is over the outcomes of the mechanism $\mathcal{M}$.

When $\delta = 0$, $\mathcal{M}$ is called (ε)-DP mechanism.

An intuition of (ε,δ)-DP is as follows: if we choose two consecutive databases (that differ exactly on one record), the mechanism will likely return indistinguishable values. In other words, it preserves privacy with high probability, but it is admissible for a mechanism to be out of control with negligible probability δ.

Example 1 (Laplace noise) In the central model, a standard and widely used mechanism with the (ε)-DP property is the so-called Laplace noise. A variable X has Laplace distribution with parameter λ (denoted as $X \sim \mathcal{L}(\lambda)$), if its probability density function is

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

Note that $\mathbb{E}\left(X\right) = 0$ and $\mathrm{Var}(X) = 2\lambda^2$.

Let $c(x)$ be the number of rows in x, which satisfy a given property. Note that c in the differential privacy literature is usually referred to as *count query*. Imagine that an aggregating mechanism is defined as follows: $\mathcal{M}(x) = c(x) + \mathcal{L}(\varepsilon^{-1})$. Then $\mathcal{M}$ is (ε)-DP (for more precise properties of Laplace noise, see Dwork and Roth (2014)).

In the privacy analysis of large-scale distributed protocols, two types of approaches are typically distinguished: event/record-level (e.g., Wang et al. (2019); Papernot et al. (2017)), where information about a specific event/record is protected, and user-level privacy (e.g., McMahan et al. (2018)), which protects the privacy of the users themselves. Generally, the latter is a stronger model (see the discussion in McMahan et al. (2018)).

Fundamentally, the model analysed in our work pertains to event-level privacy. That is, we protect information regarding whether a counter incrementation event occurred. This corresponds to a scenario in which a participant can perform a certain action only once (for example, liking a specific message on typical social media platforms). These results can be directly adapted to scenarios where a single user may be associated with multiple counter increments using standard methods. Naturally, the strength of the privacy guarantee in such a framework must depend heavily on additional assumptions (such as constraints on the number of events or the associations between users).

4 Probabilistic Counters — preliminaries

This paper focuses on *probabilistic counters*, further denoted by M. The notion of a probabilistic counter is ambiguous in the literature. It is a stochastic process that can be interpreted as a mechanism defined on the space of all possible inputs, which should estimate some goal value in some sense. The exact definition of this approximation is not crucial from the DP-point of view; thus we do not consider it in this paper.

Each increase in the data source counted by the probabilistic counter is called an *increment request*. Due to the randomised nature of probabilistic counters, each may change the value of the counter, but not necessarily. We will also indicate the single increment request by $'1'$. For the sake of generality, we also assume that the counter can get as an input $'0'$, and in such a case it simply does nothing. This is useful for real-life scenarios, e.g., data aggregation (see Section 5). Obviously, only increment requests impact the counter's final result; hence, we indicate the counter's value after n increment requests by M_n, and we are not considering the number of the rest of the rows.

In Figure 1, one can see a graphical representation of the probabilistic counter. As mentioned, increment requests are indicated by $'1'$ and other rows by $'0'$ input. The dice represent randomness. The $\mathbf{X}$-mark indicates that there is no action.

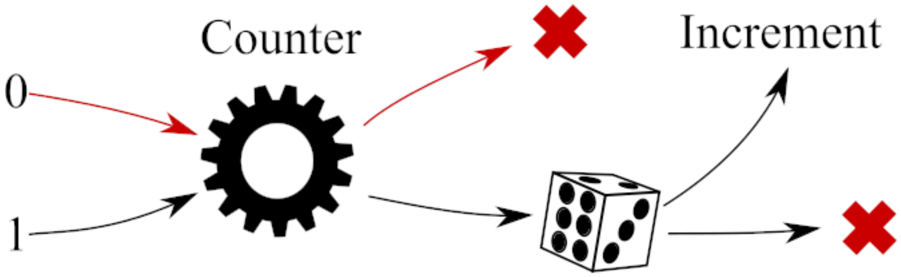

Fig. 1: Graphical depiction of the probabilistic counter.

We emphasise that the probabilistic counter depends on the number of increment requests. We want to show that if we reveal its final value, then it does not expose any sensitive data about any single record. Moreover, note that if x and y differ only by one input $'0'$, then $\mathbb{P}\left(M(x) \in \mathcal{S}\right) = \mathbb{P}\left(M_n \in \mathcal{S}\right) = \mathbb{P}\left(M(y) \in \mathcal{S}\right)$, where n is the number of increment requests for both x and y. See that then the condition in Definition 2 is trivially fulfilled. Hence, for our convenience, in this paper, we use the marking of only the number n of increment requests provided by individuals when talking about the probabilistic counter M_n.

Fact 1 *Let M be a probabilistic counter with a discrete $Range(M) = A$. Moreover assume that for all $n, m \geqslant 1$, such that $|n-m| \leqslant 1$, there exists such $S_n \subset A$ that for all $s \in S_n$*

$$\mathbb{P}\left(M_n = s\right) \leqslant \exp(\varepsilon) \cdot \mathbb{P}\left(M_m = s\right) \tag{1}$$

and

$$\mathbb{P}\left(M_n \notin S_n\right) \leqslant \delta\ . \tag{2}$$

Then M is (ε, δ)-DP.

Note that, for our setting, Fact 1 is fully compatible with the intuition of regular differential privacy (Definition 2). Indeed, Fact 1 can easily be derived from the observation that any set $B \subset A$ is a disjoint union of $B \cap S_n$ and $B \cap S_n'$.

Remark that ε and δ in Fact 1 can also be treated as functions of a parameter n, i.e. with respect to the number of increment requests. Thence, we can consider the differential privacy of the variable M_n, when n is known, or $(\varepsilon(n), \delta(n))$-DP of the probabilistic counter M. The second variant lets us provide a precise dependence of privacy parameters of the counter as the number of increment requests gets large.

4.1 Morris Counter

We begin with a short description of the Morris Counter (originally referred to as an approximate counter Morris (1978); Flajolet (1985)). Fix $a > 1$. The algorithm 1 is a very simple pseudocode of the Morris Counter Morris (1978).

```
$M \leftarrow 1$;
while receive request do
    generate $r \sim \mathrm{Uni}([0,1])$;
    if $r < a^{-M}$ then
        $M \leftarrow M + 1$;
```

Algorithm 1: Morris Counter Algorithm

Roughly speaking, we start with $M = 1$. Each incoming increment request triggers a random event. This event increments the counter ($M \leftarrow M + 1$) with probability a^{-M} ($r \sim \mathrm{Uni}([0,1])$ generates a number uniformly at random from the interval $[0, 1]$, using in practice some Pseudo Random Number Generator). Note that this approximate counting protocol can be easily distributed. Indeed, any entity who wants to increment the counter only has to send the request to increment it. These requests can be queued on the server and resolved one after another. A detailed description of the approximate counting method can be found in Morris (1978); Flajolet (1985). Throughout this article, we examine only a standard Morris Counter i.e., with the base $a = 2$. Morris Counter can also be defined recursively.

Definition 3 *The Morris Counter is a Markov process* $(M_n, n \in \mathbb{N}_0)$ *that satisfies:*

$$\begin{aligned}
&\mathbb{P}(M_0 = 1) = 1 \ , \\
&\mathbb{P}(M_{n+1} = l | M_n = l) = 1 - 2^{-l} \ , \\
&\mathbb{P}(M_{n+1} = l + 1 | M_n = l) = 2^{-l} \ ,
\end{aligned}$$

for any $l \in \mathbb{N}$ *and* $n \in \mathbb{N}_0$.

Note that Definition 3 can be derived directly from a run of Algorithm 1. From now on, let $\mathbb{P}(M_n = l) = p_{n,l}$. Directly from Definition 3 we get the following recursion:

$$p_{n+1,l} = (1 - 2^{-l})p_{n,l} + 2^{-l+1}p_{n,l-1} \tag{3}$$

for $l \in \mathbb{N}$ and $n \in \mathbb{N}_0$ with starting and boundary conditions $p_{0,1} = 1$, $p_{0,l} = 0$ for $l \geqslant 2$ and $p_{n,0} = 0$ for $n \in \mathbb{N}_0$.

Accuracy versus Differential Privacy The accuracy of Morris Counter has been thoroughly analysed in various classical papers. The first detailed analysis was proposed by Ph. Flajolet in Flajolet (1985). In this part, we present the essence of theorems presented in this paper, which will be useful later on.

First, we provide the asymptotics of the expected value and the variance of Morris Counter, with precise numerical approximations of constants:

Fact 2 *Let M_n denote Morris Counter after n successive increment requests. Then this random variable has an expected value $\mathbb{E}\left(M_n\right) \approx \log(n) - 0.27395$ (in this paper $\log$ states for binary logarithm) and a variance $\mathrm{Var}(M_n) \approx 0.763014$.*

Realize that Fact 2 guarantees high concentration of M_n around its average — a characteristic desirable in order to satisfy differential privacy definition. Fact 2 also justifies a definition of moving discrete intervals:

$$I_n = [\lceil\log(n)\rceil - 4 : \lceil\log(n)\rceil + 4] \cap [n+1] \, , \tag{4}$$

which will emerge as a crucial point of our further considerations of this Markov process in terms of differential privacy in this section. Let us mention that usually $[\lceil\log(n)\rceil - 4 : \lceil\log(n)\rceil + 4] \subseteq [n+1]$, so one may think that I_n are symmetric discrete intervals of length 8, centred at $\lceil\log(n)\rceil$.

The lion's share of applications of Morris Counter is based on counting a number of occurrences, that is, the number of increment requests. In order to estimate this value, we may use (3) and simply obtain $\mathbb{E}\left(2^{M_{n+1}}\right) = \mathbb{E}\left(2^{M_n}\right) + 1$, so together with the assumption $M_0 = 1$ we obtain the following.

$$\mathbb{E}\left(2^{M_n}\right) = n + 2 \, . \tag{5}$$

Hence $2^{M_n} - 2$ is an unbiased estimator of the number of increments n. Remark that n can be saved in $\lceil\log(n)\rceil$ bits. On the other hand, Fact 2 shows that on average, $\log(\log(n)) + O(1)$ bits are required to store M_n. As announced earlier, this is the crucial advantage of Morris Counter. Moreover, analogously to (5) we may obtain

$$\mathrm{Var}(2^{M_n} - 2) = \frac{n(n+1)}{2} \, . \tag{6}$$

Formulas (5) and (6) will be used in the example of data aggregation analysis in Section 5.

4.2 MaxGeo Counter

We begin with a short description of MaxGeo Counter. Algorithm 2 shows its pseudocode. Informally, for each increment request, the server has to generate a random variable from the geometric distribution $\mathrm{Geo}(1/2)$ (ranged in $\mathbb{N}$). The final result is the maximum taken over all these random variables generated.

```
$C \leftarrow 1$;
while receive request do
    generate $r \sim \mathrm{Geo}(1/2)$;
    $C \leftarrow \max\{C, r\}$;
return $C$;
```

Algorithm 2: MaxGeo Counter Algorithm

The expectation and variance of the maximum of n i.i.d. geometric variables have already been analysed in the literature. For instance, Szpankowski and Rego Szpankowski and Rego (1990) provided exact formulas for the' expected value and variance of such variables. However, they are impractical for large applications n. Hence, they also provided asymptotics (here, for a maximum of n independent $\mathrm{Geo}(1/2)$

distributions): $\mathbb{E}\left(M_n\right) = \log(n) + O(1)$ and $\mathrm{Var}(M_n) = \log(n) + O(1)$ and thus, similarly to the Morris Counter, there are only $\log(\log(n)) + O(1)$ bits required on average to save the MaxGeo Counter after n increment requests.

4.3 General Probabilistic Counting with Stochastic Averaging

Here we recall briefly a General Probabilistic Counting with Stochastic Averaging algorithm, based on the original idea from Flajolet and Martin (1985). Assume that there are m, initially empty lots related to some independent copies of some probabilistic counter. For each increment request, we connect it to one of the groups uniformly at random. Finally, we perform incrementation requests separately and independently for each lot, obtaining the following. $M[1], M[2], \ldots, M[m]$.

Without delving into details, for the original PCSA algorithm, $\mathbb{E}\left(M_n\right) \approx \log(\varphi n)$, where M_n is a value of a specific probabilistic counter connected with PCSA after n increment requests and φ is some magic constant. If we denote the mean of these counters m after the total number of increment requests n by $\sigma_n(m)$, then we may introduce the statistic:

$$\Xi_n(m) = \left\lfloor \frac{m}{\varphi} 2^{\sigma_n(m)} \right\rfloor .$$

Then (according to Flajolet and Martin (1985)), for any $m = 2^k$, $k \in \mathbb{N}$, $\mathbb{E}\left(\Xi_n(m)\right) \approx n\left(1 + \frac{0.31}{m}\right)$ and $\mathrm{Var}(\Xi_n(m)) = n^2\left(\frac{0.61}{m}\right)$.

Note that averaging reduces the variance of the probabilistic counter. Remark that "Stochastic Averaging" in PCSA algorithm refers to the random choice of the number of entities in each group, and it slightly differs from the standard averaging solution via the Monte Carlo method with groups of equal size.

An important conclusion is that we may apply the idea of original PCSA in general to any probabilistic counters.

4.4 LogLog counter

In Durand and Flajolet (2003) a LogLog algorithm was proposed. It is based on $m = 2^k$ counters $(M[j])_{j=1}^{m}$, where $k > 0$. We may interpret this algorithm in the context of probabilistic counters. In such a scenario, it takes a hashed value (binary sequence) as input on every increment request. The first k bits of the hash determine which of the m counters should be incremented (the index j is chosen as the decimal representation of the sequence restricted to these first k bits; hence it translates the increment request of the LogLog counter to the increment request of one of the m internal counters). Consider the first non-zero bit of a tail of the sequence (starting from $(k+1)$-th bit). Its position R in this tail follows the $\mathrm{Geo}\left(\frac{1}{2}\right)$ distribution assuming the uniform distribution of the input sequences. If $R > M[j]$, then $M[j]$ should become R. Otherwise, it does not change.

Therefore, LogLog counter is, in fact, a general PCSA that uses m MaxGeo counters. It can be used to estimate the cardinality of increment requests n using the following estimator:

$$\mathrm{LogLog}_n^{(m)} = \alpha_m m 2^{\frac{1}{m}\sum\limits_{j=1}^{m} M_n[j]} ,$$

with the scaling constant given by the formula

$$\alpha_m = \left(\Gamma\left(-\frac{1}{m}\right)\frac{2^{-1/m} - 1}{\ln(2)}\right)^{-m} ,$$

where Γ is Euler's Gamma function.

It is worth mentioning that α_m is an increasing sequence and $\alpha_m \approx 0.79$ for $m \geqslant 64$ (e.g. $\alpha_8 \approx 0.69763\ldots$, $\alpha_{64} = 0.78356\ldots$ and the limit is $\alpha_\infty = 0.79402\ldots$).

We are interested in the expectation and accuracy of the cardinality estimator, which can be briefly described as follows: $\mathbb{E}\left(\mathrm{LogLog}_n^{(m)}\right) \approx n$ and $\mathrm{Var}(\mathrm{LogLog}_n^{(m)}) \approx \frac{1.69n^2}{m}$ (see Durand and Flajolet (2003) for more details). However, in order to control the LogLog counter, on average about $m \log\log\left(\frac{n}{m}\right)$ bits of memory are needed.

An interesting fact is that the expectation of a single MaxGeo counter is logarithmic and the estimator of n is of the form $Cm2^{\sigma_n(m)}$, for some constant C (just as in the case of the original PCSA).

Since LogLog counter is an effect of processing of some MaxGeo counters, therefore its differential privacy is based on the same property of the auxiliary counters.

4.5 HyperLogLog

The maximum of geometric variables is also used as a primitive in the well known HyperLogLog algorithm (see Flajolet et al. (2007)). Therefore, its privacy properties are important both from the theoretical and practical point of view. Essentially, in HyperLogLog we perform the general PCSA algorithm, but the final estimation is somehow different:

$$\mathrm{HyperLogLog}_n^{(m)} := \alpha_m m^2 \left(\sum_{j=1}^{m} 2^{-M_n[j]}\right)^{-1},$$

where α_k is a constant dependent only on k (see Flajolet et al. (2007) for more details). It should be noted that HyperLogLog related algorithms (mentioned in Section 1) are the best-known procedures designated for cardinality estimation, and are close to optimum Indyk and Woodruff (2003). According to Flajolet et al. (2007), for $m = 2^k$, where $k \geqslant 4$,

$$\mathbb{E}\left(\mathrm{HyperLogLog}_n^{(m)}\right) = n(1 + \psi_3(n) + o(1)), \text{ with } |\psi_3(n)| < 5 \cdot 10^{-5}$$

and

$$\mathrm{Var}(\mathrm{HyperLogLog})_{\mathrm{n}}^{(\mathrm{m})} = n^2 \left(\frac{\beta_m}{\sqrt{m}} + \psi_4(n) + o(1)\right)^2, \text{ with } |\psi_4(n)| < 5 \cdot 10^{-4},$$

where $\beta_m \overset{m\to\infty}{\longrightarrow} \sqrt{2\log(2) - 1} = 1.03896\ldots$ and $\beta_m \leqslant 1.106$ for $m \geqslant 16$.

5 Privacy-Preserving Survey via Probabilistic Counters

In this section, we present an example scenario for data aggregation using probabilistic counters. We assume that there is a *server* (alternatively, we call it *aggregator*) and a collection of *nodes* (e.g., mobile phone users), and we want to perform a boolean survey with a sensitive question. That is, each user sends $'0'$ if his answer is *no* and $'1'$ if the answer is *yes*. We assume that the connections between users and the server are perfectly secure, and the data can safely get to the trusted server. This can be performed using standard cryptographic solutions. The server's goal is to publish the sum of all responses $'1'$ in a way that preserves privacy. This goal could obviously be achieved by simply collecting all the data and adding an appropriately calibrated Laplace noise (see Dwork and Roth (2014)). However, we aim to show that probabilistic counters have inherently sufficient randomness to be differentially private without any auxiliary randomising mechanism.

We can present the general scenario in the following way:

1. each user sends his/her bit of data to the server using secure channels,
2. server plugs the data points sequentially into the counter,
3. if the data point is $'1'$, the counter receives *increment request*, otherwise, the data is ignored,
4. each increment request is being processed by the counter and may lead (depending on randomness) to an increase of the value of the counter,
5. when all data are processed, the value of the counter is *released* to the public.

Note that we assume that the Adversary has access **only** to the released value. We also released only the counter value itself, which does not estimate the responses of $'1'$. This estimation is a function of the released value, which is different for Morris or MaxGeo Counter. There can also be various ways to estimate the actual number using a counter value. However, this does not matter for our case, as differential privacy is conveniently fully resilient to post-processing (see Dwork and Roth (2014)). The graphical representation of our scenario considered is presented in Figure 2.

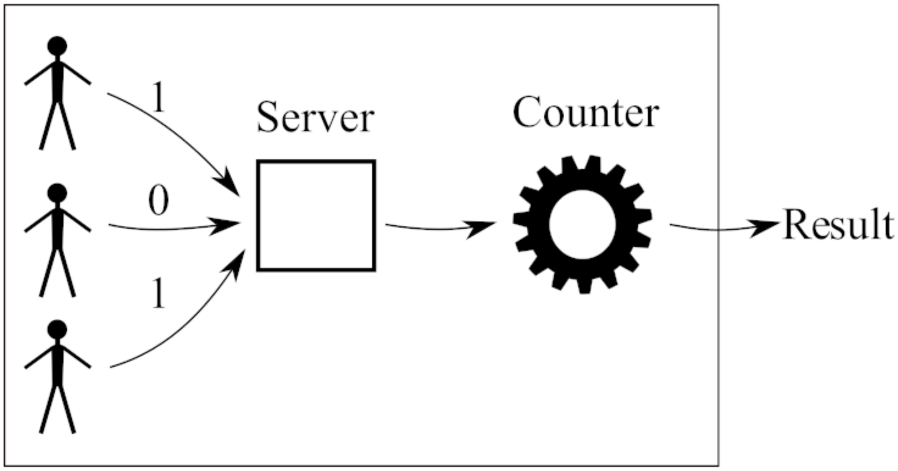

Fig. 2: Scenario for data aggregation using probabilistic counters. We assume that the Adversary does not have any way to extract information from within the rectangle.

Adversary Our assumptions about the Adversary are the same as in most differential privacy papers. Namely, he may collude with any subset of the participants (e.g., all except the single user whose privacy he wants to breach). On the other hand, the aggregator is trusted. See that even though we have a distributed system in mind, this is, in fact, a central differential privacy scenario. We do not assume pan-privacy. This means that the algorithm's internal state is **not** subject to the constraints of differential privacy. Obviously, if the Adversary knew the internal state of the counter at any time or could observe whether, after receiving data from a specific user, the server had to perform computations to potentially increment the counter (implying a $'1'$ response) or not, he would easily violate the privacy. We also do not assume privacy under continual observation. The survey is not published iteratively but once only after it is finished. In short, the Adversary **cannot**

- extract or tamper with the internal state of the counter,
- extracts any information from the server or channels between users and the server.

The Adversary **can**

- collude with any subset C of the participants (e.g., know their data or send them all $'0'$ to the server) in order to breach the privacy of the user not belonging to C,
- obtain the final result of the aggregation and perform any desired post-processing on it.

6 Formulation of main results

In this section, we state main results. The proofs are postponed to Section 8 for the convenience of the reader.

6.1 Morris Counter Privacy

In this subsection, we investigate the Morris Counter in terms of (ε, δ)-DP. Here we present the main result:

Theorem 1 *Let M denote the Morris Counter and assume $|n-m| \leqslant 1$. Then*

$$\mathbb{P}(M_n = l) \leqslant \left(1 - \frac{16}{n}\right)^{-1} \cdot \mathbb{P}(M_m = l) + \delta,$$

where $\delta < 0.00033$, so M is $(L(n), 0.00033)$-DP with

$$L(n) = -\ln\left(1 - \frac{16}{n}\right) = \frac{16}{n} + \frac{128}{n^2} + O(n^{-3}) \leqslant \frac{16}{n-8}\ .$$

An explanation is postponed to subsection 8.1.

6.2 General result on Morris' Counter privacy

In this part, we show that the Morris' Counter guarantees privacy with both parameters tending fast to zero. The analysis is based on the observations from the previous case. However, instead of I_n (see Section 4.1 for the discussion), we consider intervals

$$J_n(c) = [\lceil \log(n) \rceil - \lceil c \log(\ln(n)) \rceil : \lceil \log(n) \rceil + \lceil c \log(\ln(n)) \rceil] \cap [n+1]\ ,$$

where c is some positive constant such that $\lceil c \log(\ln(n)) \rceil \geqslant 1$, for large enough n.

We can now state our next contribution:

Theorem 2 *Let M denote the Morris Counter. If $c > 0$ satisfies $\lceil c\log(\ln(n))\rceil \geqslant 1$, then M is $(\varepsilon(n), \delta(n))$-DP with parameters $\varepsilon(n) = O\left(\frac{(\log(n))^2}{n}\right)$ and $\delta(n) = O\left(n^{-(\ln(n))^{c-1}} + n^{-1}(\ln(n))^{-c}\right)$.*

A proof is given in subsection 8.2

6.3 MaxGeo Counter Privacy

In this subsection, we present a theorem that shows the privacy guarantees of MaxGeo Counter. Assume that we have n increment requests. In the case of MaxGeo Counter, it means that we generate random variables $X_1, \ldots, X_n$, where $X_i \sim \mathrm{Geo}(1/2)$ are pairwise independent. Ultimately, the result of the counter is maximum over all X_i's, namely $X = \max(X_1, \ldots, X_n)$. Now we are ready to present our second main contribution.

Theorem 3 *Let M denote the MaxGeo Counter, and n denote the number of increment requests. Consider m such that $|n - m| \leqslant 1$. Fix $\varepsilon > 0$ and $\delta \in (0, 1)$ and let*

$$l_\varepsilon = \left\lceil \log\left(\frac{e^\varepsilon}{e^\varepsilon - 1}\right)\right\rceil .$$

If

$$n \geqslant \frac{\ln(\delta)}{\ln\left(1 - 2^{-l_\varepsilon}\right)} \left(\approx -\frac{\ln(\delta)}{\varepsilon}\right) , \tag{7}$$

then

$$\mathbb{P}\left(M_n \in S\right) \leqslant \exp(\varepsilon) \cdot \mathbb{P}\left(M_m \in S\right) + \delta,$$

so M is (ε, δ)-DP.

A proof is postponed to subsection 8.3.

7 Practical applications

7.1 Discussion on arragements of the survey scenario

We usually consider probabilistic counters to be some kind of estimator. There is always a trade-off between its precision and the quality of privacy. Ultimately, it means that for fixed privacy parameters (ε, δ) we can calculate the minimum number of increment requests necessary to satisfy given privacy parameters. This can be done by artificially adding them before actually collecting data. Of course, it has to be taken into account that the initial added value should be subtracted from the final estimation of the appropriate cardinality before publication, and this change can impact the precision of the estimation (especially when the expected number of increment requests is very small). If we can perform such a preprocessing, then for every (ε, δ), we can easily know how many artificial counts have to be added. Nevertheless, we have to be aware that if the total number of increment requests is small, we may obtain a poor approximation, so this approach should be used whenever the privacy is much more important than the precision.

Note that, in light of our theorems 1 and 3, both the Morris Counter and the MaxGeo Counter preserve differential privacy in such a scenario. Assume that at least n users have $'1'$, therefore at least n incrementation requests. See that we can either know it based on domain knowledge (e.g., we expect that at least some fraction of users will send $'1'$ based on similar surveys) or add x counts to the counter artificially initially. The number x should be chosen according to the maximal amenable value of the parameter ε for a given application, but we recommend choosing rather small values of x. Obviously, in the case of artificial counts, this has to be taken into account when estimating the final sum. Using the Morris counter, we obtain $(L(n), 0.00033)$-DP with

$$L(n) = -\ln\left(1 - \frac{16}{n}\right) \leqslant \frac{16}{n-8}\ .$$

Example 2 Consider a result of Morris Counter with a small number n of increment requests (for example, a number of respondents suffering from a rare illness). Therefore, we will likely require the parameter ε to be at most some threshold, e.g. 1. Therefore, from Theorem 1, we should add x counts where $L(x) \leqslant 16/(x-8) \leqslant 1$, so $x \geqslant 24$. Note that we do not include n in the above formula since it is not known in advance. Therefore, using the Morris Counter, the above survey is at least $(L(n+24), 0.00033)$-DP. However, the estimator should be modified as well, i.e., $M'_n = M_{n+24}$, so $\hat{n'} = \max\{2^{M'_n} - 26, 0\}$ (since $2^{M'_n}$ may be smaller than 26).

On the other hand, using the MaxGeo Counter for a given ε and δ we get (ε, δ)-DP as long as $n \geqslant \dfrac{\ln(\delta)}{\ln(1 - 2^{-l_\varepsilon})}$, where $l_\varepsilon = \lceil \log(1 + 1/\varepsilon) \rceil$ (see Theorem 3).

Example 3 Assume that we have at least $n = 200$ increment requests. From Theorem 1, we have $L(n) \leqslant 16/(n-8) \leqslant 0.08334$. Hence, using the Morris Counter, the above survey is $(0.08334, 0.00033)$-DP.

On the other hand, using the MaxGeo Counter for a given ε and δ we get (ε, δ)-DP as long as $n \geqslant \dfrac{\ln(\delta)}{\ln(1 - 2^{-l_\varepsilon})}$, where $l_\varepsilon = \lceil \log(1 + 1/\varepsilon) \rceil$.

Example 4 Let $\varepsilon = 0.5$ and $\delta = 1/D^2$, where D is the number of all the survey participants. After using our theorem and straightforward calculations, we have $n \geqslant 7\ln(D)$. Say we will have $\lfloor \exp(20) \rfloor$ participants. Then if we have at least 140 increment requests, we satisfy $(0.5, 1/D^2)$-DP.

Note that from a differential privacy perspective, both the general PCSA algorithm and HyperLogLog can be seen as arbitrary postprocessing performed on the m MaxGeo counters. Moreover, since each response goes to one counter only, they are independent of each other, so we can use the parallel composition theorem (see Dwork and Roth (2014)).

Observation 1 *Assume we have k MaxGeo Counters $M[1], \ldots, M[m]$, which are used either in HyperLogLog or PCSA algorithm. If jth MaxGeo Counter is $(\varepsilon_j, \delta_j)$-DP then the chosen algorithm is $(\max_i \varepsilon_i, \max_i \delta_i)$-DP.*

7.2 Comparison of Morris and MaxGeo Counters

In this subsection, we compare the' privacy and storage properties of data aggregation algorithms based on one of the investigated counters or the standard Laplace method.

We start with auxiliary remarks for the privacy of MaxGeo Counter. For instance, see that if δ and n

are fixed, then from Theorem 3 and $l_\varepsilon \leqslant \left\lceil \ln\left(1+\varepsilon^{-1}\right)\right\rceil$ we obtain

$$\varepsilon(n) \geqslant \left(2^{\left\lfloor -\log\left(1-\delta^{\frac{1}{n}}\right)\right\rfloor} - 1\right)^{-1} =: \varepsilon_0(n)\ . \tag{8}$$

We want to optimise $\varepsilon(n)$, so we will consider $\varepsilon_0(n)$ defined as the right-hand side of (8). In order to limit it let us consider the following function of $x \in \mathbf{R}_+$:

$$\psi(x,\delta) := \left(\left(1-\delta^{\frac{1}{x}}\right)^{-1} - 1\right)^{-1} = -\frac{\ln(\delta)}{x} + \frac{\ln(\delta)^2}{2x^2} - \frac{\ln(\delta)^3}{6x^3} + O(x^{-4})\ . \tag{9}$$

Naturally, then $\varepsilon_0(n) \geqslant \psi(n,\delta) = -\ln(\delta)/n + O(n^{-2})$. Since ψ is decreasing with respect to x, we will consider when $\varepsilon_0(n)$ changes. More precisely, consider a minimal k such that $\varepsilon_0(n) < \psi(n-k,\delta) \leqslant \varepsilon_0(n-k)$, which appears to be the neat upper bound for $\varepsilon(n)$. However, since $\varepsilon_0(n)$ is the non-ascending step function, we realise that

$$\varepsilon_0(n-k) \geqslant \left(2^{\left\lfloor -\log\left(1-\delta^{\frac{1}{n}}\right)\right\rfloor -1} - 1\right)^{-1} = -\frac{2\ln(\delta)}{n} + \frac{3\ln(\delta)^2}{n^2} - \frac{13\ln(\delta)^2}{3n^3} + O(n^{-4})\ .$$

If we denote $\phi(n,\delta) := \left(2^{\left\lfloor -\log\left(1-\delta^{\frac{1}{n}}\right)\right\rfloor -1} - 1\right)^{-1}$, then we can summarise our recent considerations in a short way by $\psi(n,\delta) \leqslant \varepsilon(n) < \phi(n,\delta)$. Thence, in the case where we fix the parameter $\delta = 0.00033$, we obtain

$$\frac{8.0164\ldots}{n} + \frac{32.13147\ldots}{n^2} + O(n^{-3}) \leqslant \varepsilon(n) \leqslant \frac{16.0328\ldots}{n} + \frac{192.789\ldots}{n^2} + O(n^{-3})\ .$$

On the other hand, from Theorem 1, we know that when $\delta = 0.00033$, then for Morris Counter (with $\varepsilon(n)$ defined by (10)) the quite similar relation holds:

$$\varepsilon(n) \leqslant -\ln\left(1-\frac{16}{n}\right) = \frac{16}{n} + \frac{128}{n^2} + O(n^{-3})\ .$$

Therefore, Morris and MaxGeo Counters behave quite similarly under comparable conditions, and Figure 3 confirms this observation. Indeed, in Figure 3 we can see that the difference between the values of the parameters $\varepsilon(n)$ for both counters decreases as n increases.

Realise that the previous conclusions remain true if $\delta(n)$ is not constant. This short observation enables us to obtain a more general result:

Fact 3 *Let $\delta(n) = n^{-c}$ for some constant $c > 0$. Then*

$$\varepsilon(n) \leqslant \phi(n,\delta(n)) = \frac{2c\ln(n)}{n} + \frac{3c^2\ln(n)^2}{n^2} + O\left(\frac{c^3\ln(n)^3}{n^3}\right)$$

and MaxGeo Counter is $(\phi(n,\delta(n)),\delta(n))$-DP for any $n \in \mathbb{N}$.

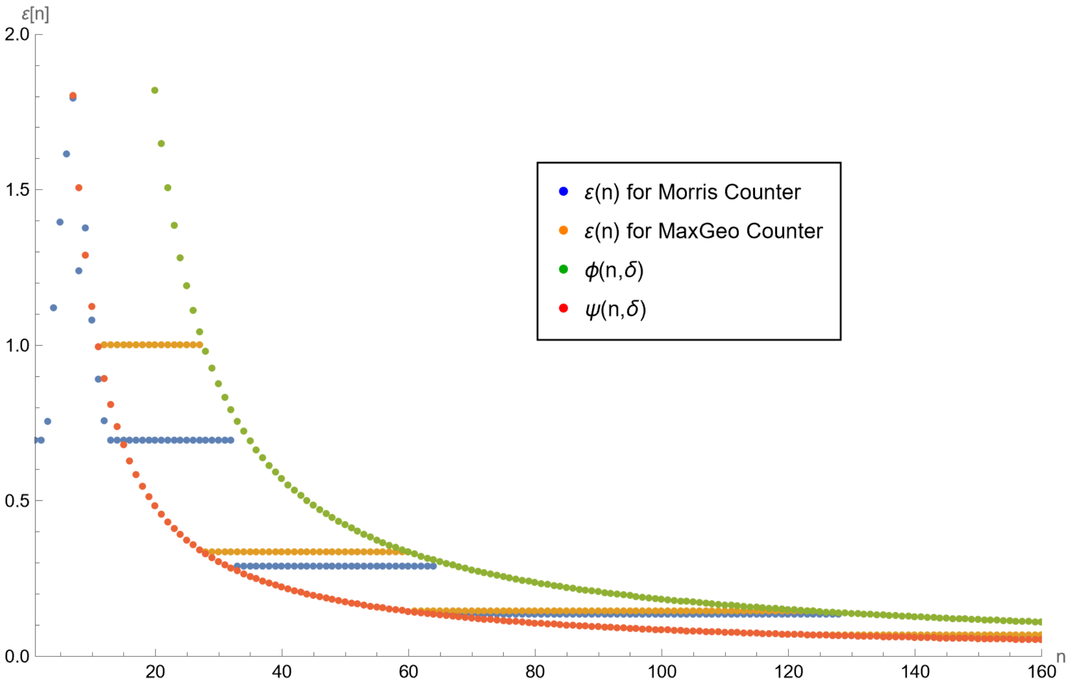

Fig. 3: Values of $\varepsilon(n)$ parameters for Morris and MaxGeo Counters compared with boundaries for $\varepsilon(n)$ for MaxGeo Counter: the lower one — $\psi(n,\delta)$ and the upper one — $\phi(n,\delta)$ ($n \leqslant 160$ and $\delta = 0.00033$).

Notice that in this case, both sequences of parameters tend to 0, which may be used as an advantage in applications, especially when we expect that the total number of increment requests will be very large. However, we emphasise that this requires that $\delta(n)$ be negligible.

In Figure 4, we may briefly see that probabilistic counters can be used for data aggregation to decrease memory usage in exchange for a slight increase of the parameter δ of differential privacy and wider confidence intervals (lower accuracy). In recent years Big Data related problems became very popular. Note that this kind of application makes major use of memory. When the server aggregates many different data, standard solutions may cause a serious problem with data storage, which can be encountered by using the idea based on a probabilistic counter instead.

Example 5 Imagine that 100 million people participate in a general health survey with 100 *yes/no* sensitive questions.For every question, we would like to estimate the number of people who answered *yes*, but we want to guarantee the differential privacy property at a reasonable level. Realise that if the number of *yes* answers is very small for some questions (e.g., when the question is about a very rare disease), then the number of *no* answers may be counted instead. Obviously, this method gains privacy, but loses precision.

According to Figure 4, if we use the Laplace method, then we may need approximately $100 \log(10^8) = 2657.54\ldots$ bits to store the counters. Let us note however, if we use the Morris Counter instead, about $100 \log(\log(10^8)) = 473.20\ldots$ bits are needed. Note that all terms $O(1)$ in the "Average memory" row

Method	Laplace noise $\mathcal{L}(n/16)$	Morris counter M_n	LogLog$^{(2)}$ counter $(M_n[1], M_n[2])$
(ε, δ)-DP	$(16/n, 0)$-DP	$(\frac{16}{n-8}, \delta)$-DP	$(\sim \frac{32.066}{n}, \delta)$-DP
Estimator $\hat{n}$	$n + \mathcal{L}(n/16)$	$2^{M_n} - 2$	$\sim 0.89 \cdot 2^{\frac{1}{2}\sum_{i=1}^{2} M_n[i]}$
$\mathrm{Var}(\hat{n})$	$\frac{n^2}{128}$	$\frac{n^2+n}{2}$	$\sim 0.845 \cdot n^2$
Avg. memory	$\log(n) + O(1)$	$\log(\log(n)) + O(1)$	$2\log(\log(\frac{n}{2})) + O(1)$

Fig. 4: A summary of data aggregation techniques. The standard one is based on the Laplace method, and the rest are based on probabilistic counters. Recall that $\delta = 0.00033$.

of Figure 4 are bounded by 1. Hence, its impact is negligible from a practical point of view.

One may also complain about the heavy use of the pseudo random number generator (PRNG) that probabilistic counters make. However, this problem may be resolved by generating the number of increment requests which have to be forgotten until the next update of the counter by using appropriate geometric distributions (see, for example, Li (1994) for a similar approach applied to the reservoir sampling algorithm). This way, the use of PRNG can be substantially reduced.

8 Proofs

8.1 Main Theorem for Morris Counter (Theorem 1)

The proof is complicated and very technical. In order to better understand it, we are going to provide a presentation of a plan and main ideas beneath the parts of the proof. Let $n_k := 2^k + 1$ for $k \in \mathbb{N}$.

We introduce $\mathcal{P}_k^{(c)}$ as the probability that the Morris Counter M_n being $k + c$ after n_k increment requests, i.e., $p_{n_k, k+c}$. The "special" sequences $(\mathcal{P}_k^{(c)})_k$ play a crucial role in the proof.

Roadmap of the proof: We can divide the proof of Theorem 1 into five phases (the main results of the phases are given in brackets):

1. δ phase (Theorem 5),
2. relations between "special" sequences $(\mathcal{P}_k^{(c)})_k$ with respect to c (Claim 1),
3. dependencies between consecutive distributions (of M_n and M_{n+1}) (Claim 2),
4. extrapolation of $\mathcal{P}_k^{(c)} \leq 2^{c+3}\mathcal{P}_k^{(c+1)}$ property to $N > n_k$ (Lemma 3),
5. ε phase (Theorem 6).

During the first phase, we consider a concentration of Morris Counter in the vicinity of its mean value. More precisely, we show that the Morris Counter after n increment requests takes values in relatively small intervals I_n with probability $1 - \delta$ (note that then M_n satisfies condition (2) for $S_n = I_n$), where I_n is defined as in (4) and δ is some small constant, which arises from the proof. Note that I_n can be interpreted as confidence intervals at level $1 - \delta$ (see, e.g. Neyman (1937)). This phase is divided into lemmas 1 and 2. The first one uniformly bounds the formula for probabilities given by the Theorem below, due to Flajolet:

Theorem 4 (Proposition 1 from Flajolet (1985)) *The probability $p_{n,l}$ that the Morris Counter has value l after n increment requests is*

$$p_{n,l} = \sum_{j=0}^{l-1} (-1)^j \, 2^{-j(j-1)/2} \left(1 - 2^{-(l-j)}\right)^n \prod_{i=1}^{j} \left(1 - 2^{-i}\right)^{-1} \prod_{i=1}^{l-1-j} \left(1 - 2^{-i}\right)^{-1} .$$

We sum up the bounds on $p_{n,l}$ to obtain a small upper bound for $\delta_1 := \mathbb{P}\left(M_n \leqslant \lceil \log(n) \rceil - 5\right)$ (Lemma 1). The same bounds cannot be utilised efficiently in the proof of Lemma 2. Instead, it couples M_n with a process X_n, which increases during the first $\lceil \log(n) \rceil + 1$ steps, and then follows the same update rule, so $M_n \leqslant X_n$ almost surely.

Therefore $\delta_2 := \mathbb{P}\left(M_n \geqslant \lceil \log(n) \rceil + 5\right) \leqslant \mathbb{P}\left(X_n \geqslant \lceil \log(n) \rceil + 5\right)$, which is much easier to bound from definition. Note that such a coupling cannot be used in the proof of Lemma 1. The first phase is summarised by Theorem 5, i.e., establishes $\delta = \delta_1 + \delta_2$.

In the second phase, we show that $(\mathcal{P}_k^{(4)})_k$ is descending for large enough k and $(\mathcal{P}_k^{(5)})_k$ is ascending for big enough k (Lemma 6). A change of monotonicity is the first obstacle in the proof of the main theorem. The main idea beneath the proof of this fact is to calculate the differences between consecutive elements of the considered "special" sequences by representing them as the sums via application of Flajolet's Theorem 4 and realising that usually at most first ten terms of the sums are crucial (on the other hand, taking less than eight terms is rarely sufficient). This is the second issue, which makes the proof so complicated. Let us note that Theorem 4 presents an explicit formula for $\mathbb{P}\left(M_n = l\right)$, which (as we may experience in A) is not convenient to analyse. However, it is simple enough to find the values numerically (also note that recursive Definition 3 provides those probabilities easily as well, However, this approach is inefficient in terms of memory and time for a large number of requests n). Therefore, by precise analysis, we can finally check some sums numerically and obtain the thesis of Lemma 6 for $k \geqslant 15$. However, numerically, one can extrapolate it to some smaller k as well (remark that this proof does not work for small values of k, since not only ten terms of the aforementioned sums are important). The lemma 6 can be used directly to show that $\mathcal{P}_k^{(4)} \leqslant 2^7 \mathcal{P}_k^{(5)}$ for $k \geqslant 7$ (Claim 1). Let us note that, in fact, this result is not true for $k < 7$.

The third phase is based on the foregoing intuition: If the ratios of consecutive probabilities of distribution of M_n increase almost exponentially, then the ratios of distribution of M_{n+1} increase similarly (but slightly slower). The proofs mainly use the definition of the Morris counter.

The fourth phase (i.e. Lemma 3) shows that if $p_{n_k,k+c} \leqslant 2^{c+3} p_{n_k,k+c+1}$, for $c \in [-k : 4]$, then the same is true if we substitute n_k with a bigger number (i.e., this property is increasing with respect to the n parameter). In order to apply this result, we have to satisfy some starting conditions. We have numerically checked the appropriate condition for $k = 7$ (presented later, in Table 1). Therefore, the second phase of the proof of Theorem 1 justifies the assumptions of Lemma 3 with respect to the parameter k, as long as $k \geqslant 7$ and the third phase let us obtain the appropriate assumptions with respect to c. One can check that for $k < 7$, an analogous assumption is not true.

The latter phase begins with the application of the result from the previous part. We obtain ε is at most $L(n)$ (provided in the formulation of Theorem 1), for $k \geqslant 7$. The last piece of this puzzle is justified by a numerical evaluation for $k < 7$ (presented later, in Figure 5), which ends the ε phase and so the whole proof.

Proof In the following, we present the proof of the main contribution. Nevertheless, some technical lemmas are given in the A.

δ **phase** Let us begin with a reminder. First, $M_n \in \mathbb{N}_0$ and, moreover,

$$I_n \subseteq [\lceil \log(n) \rceil - 4 : \lceil \log(n) \rceil + 4] \, .$$

We provide few facts about the concentration of the distribution of the random variable M_n, or more precisely about the probability that M_n will be outside the interval I_n.

Lemma 1 *Let M_n be the state of the Morris counter after the n increment requests. Then*

$$\delta_1 := \mathbb{P}\left(M_n \leqslant \lceil \log(n) \rceil - 5\right) \leqslant 0.000006515315\ldots \, .$$

An increasing sequence $\prod_{i=1}^{k} \left(1 - 2^{-i}\right)^{-1}$ that emerged in Theorem 4 will be indicated by r_k (with $r_0 = 1$) and we denote its limit $\prod_{i=1}^{\infty} \left(1 - 2^{-i}\right)^{-1} = 3.46274\ldots$ by R.
Let us mention this with the notions q_k and Q from Flajolet (1985), $r_k = \frac{1}{q_k}$ and $R = \frac{1}{Q}$.

Proof: At first, we want to bound a lower tail of the distribution δ_1.

Here we would like to find a sufficient upper limit for the above probability. Assume that $l \leqslant \lceil \log(n) \rceil - 5$.

Realise that $r_k \leqslant R$ and that $y = -x + 1$ is a tangent line to the plot of $y = -x(x-1)$ in the point $(x, y) = (1, 0)$. Therefore:

$$\begin{aligned} p_{n,l} &\overset{\text{Thm 4}}{\leqslant} \sum_{j=0}^{l-1} 2^{-\frac{j(j-1)}{2}} \left(1 - 2^{-(l-j)}\right)^n r_j r_{l-1-j} \leqslant R^2 \left(1 - 2^{-l}\right)^n \sum_{j=0}^{l-1} \sqrt{2}^{-j+1} \\ &\leqslant R^2 \frac{2}{\sqrt{2} - 1} \exp(-n2^{-l}) = R^2 (2\sqrt{2} + 2) \exp(-n2^{-l}). \end{aligned}$$

The above formula will help us limit the left tail of the distribution of M_n:

$$\begin{aligned} \delta_1 = \sum_{l=1}^{\lceil \log(n) \rceil - 5} \mathbb{P}\left(M_n = l\right) &\leqslant R^2 (2\sqrt{2} + 2) \sum_{l=1}^{\lceil \log(n) \rceil - 5} \exp(-n2^{-l}) \\ &\leqslant R^2 (2\sqrt{2} + 2) \sum_{k=4}^{\infty} \exp(-2^k) \leqslant R^2 (2\sqrt{2} + 2) \sum_{k=1}^{\infty} \exp(-16k) \\ &= R^2 (2\sqrt{2} + 2) \frac{\exp(-16)}{1 - \exp(-16)} = 0.000006515315\ldots \, . \end{aligned}$$

□

Remark The bound for $p_{n,l}$ obtained above is useless when $l \geqslant \log(n) - 2$, so it cannot be used in the next lemma for a symmetric upper tail.

Lemma 2 *Let M_n be the state of the Morris Counter after the n increment requests. Then*

$$\delta_2 := \mathbb{P}\left(M_n \geqslant \lceil \log(n) \rceil + 5\right) \leqslant 0.000325521\ldots\ .$$

Proof:

Consider a process $X = (X_{k \in [0:n]})$. Let X initially follow the incrementation rule $\mathbb{P}\left(X_k = k+1\right) = 1$ for $k \in [0 : \lceil \log n \rceil + 1]$. Afterwards, let this Markov chain imitate the transition rule of Morris Counter, that is

$$\mathbb{P}\left(X_{k+1} = m+1 | X_k = m\right) = \frac{1}{2^m} = 1 - \mathbb{P}\left(X_{k+1} = m | X_k = m\right)$$

for $k \geqslant \lceil \log(n) \rceil + 1$. Naturally, for $k \leqslant \lceil \log(n) \rceil + 1$, we have $X_k \geqslant M_k$, so we may couple realisations of these two processes in such a way that whenever X increases, so is M and if M does not change, then X does not increase as well (note that X has at most the same probability of a positive increase as M at any point in time).

To abbreviate the expressions, let us denote $m = n - \lceil \log(n) \rceil - 1$ and

$$\mu_\iota = \mathbb{P}\left(X_{k+1} = \lceil \log(n) \rceil + \iota + 1 | X_k = \lceil \log(n) \rceil + \iota\right) = \frac{1}{2^{\lceil \log(n) \rceil + \iota}} = 1 - \nu_\iota\ ,$$

for any $\iota \in \mathbb{Z}$. Moreover, let us define a three-dimensional discrete simplex:

$$S_k^{(3)} = \{\bar{l} = (l_1, l_2, l_3) \in \mathbb{N}_0^3 : l_1 + l_2 + l_3 \leqslant k\}\ .$$

Thus,

$$\begin{aligned}\delta_2 \leqslant \mathbb{P}\left(X_n \geqslant \lceil \log(n) \rceil + 5\right) &= \sum_{\bar{l} \in S_{m-3}^{(3)}} \nu_2^{l_1} \mu_2 \nu_3^{l_2} \mu_3 \nu_4^{l_3} \mu_4 \leqslant \sum_{\bar{l} \in S_{m-3}^{(3)}} \frac{1}{2^{3\lceil \log(n) \rceil + 9}} \\ &= \sum_{k=0}^{m-3} \binom{k+3}{2} \frac{1}{2^{3\lceil \log(n) \rceil + 9}} \leqslant \frac{1}{2^{10} n^3} \sum_{k=3}^{m} k^2 - k\ .\end{aligned}$$

Realise that $\sum\limits_{k=3}^{m} k = (m-2)(m+3)/2$ and $\sum\limits_{k=3}^{m} k^2 = (m-2)(2m^2 + 7m + 15)/6$, so

$$\begin{aligned}\delta_2 &\leqslant \frac{1}{2^{10} n^3} \frac{1}{6} (m-2)(2m^2 + 4m + 6) = \frac{1}{3 \cdot 2^{10} n^3} (m^3 - m - 6) \\ &\leqslant \frac{m^3}{3 \cdot 2^{10} n^3} \leqslant \frac{1}{3 \cdot 2^{10}} = 0.000325521\ldots\ .\end{aligned}$$

Note that when $m < 3$ (that is, when $n < 7$), then the above sums are empty, but on the other hand $\lceil \log(n) \rceil + 5 > n + 1$, so the inequality is trivially true. □

Theorem 5 *The state of the Morris Counter after n increment requests is **not** in the set*

$$I_n = [\lceil \log(n) \rceil - 4 : \lceil \log(n) \rceil + 4] \cap [n+1]$$

with probability $\delta < 0.00033$.

Proof: Realise that $\mathbb{P}(M_n \in [1 : n+1]) = 1$. This observation, together with the lemmas 1 and 2 yields

$$\delta := \mathbb{P}(M_n \notin I_n) = \delta_1 + \delta_2 < 0.00033\ .$$

□

The second phase In this part of the investigation, we try to establish the $\varepsilon(n)$ parameter of DP of M_n. In fact, it remains to examine the property (1) in the interval I_n, as Theorem 5 entails (2) for $S_n = I_n$. Therefore, we are interested in finding the upper bound for maximal privacy loss for any $n \in \mathbb{N}$ and $k \in I_n$, namely:

$$\varepsilon(n) = \max\left\{ \left| \ln\left(\frac{p_{n\pm1,k}}{p_{n,k}}\right) \right| : k \in I_n \right\}\ . \tag{10}$$

Actually, we may consider the sign $'+'$ instead of $'\pm'$ in (10), because $|\ln(x)| = |\ln(1/x)|$. However, when $I_n \neq I_{n\pm1}$, we have to behave carefully, so in particular, an additional cheque of privacy loss with the sign $'-'$ is needed when n is of a form $2^l + 1$ for some $l \in \mathbb{N}$.

Claim 1 *For $k \geqslant 7$, we have $p_{2^k+1,k+4} \leqslant 2^7 p_{2^k+1,k+5}$.*

The above claim is the result of a simple application of Lemma 6 from A.

The third phase

Claim 2 *If for any given n, there exists an ascending and positive sequence $(\alpha_i)_{i=1}^n$ such that*

$$(\forall\ i \in [1:n])\ p_{n,i} = 2^i \alpha_i p_{n,i+1},$$

then there also exists an ascending and positive sequence $(\alpha'_i)_{i=1}^{n+1}$ such that

$$(\forall\ i \in [1:n+1])\ (p_{n+1,i} = 2^i \alpha'_i p_{n+1,i+1}) \wedge (\forall\ i \in [1:n])\ (\alpha'_i < \alpha_i)\ .$$

This claim arises from lemmas 7 and 8 from A.

The fourth phase We use Claim 2 to guarantee starting conditions for the next Lemma 3. However, in order to apply Lemma 3, we will also use Claim 1, which assumes that $n \geqslant 2^7 + 1$. Hence, we would like to gather some information about the distribution of M_{2^7+1}. More precisely, we are interested in the behaviour of $\theta_i = p_{129,i}/p_{129,i+1}$ for $i \leqslant 11$, which we present in Table 1. We briefly see a superexponential trend of proportions θ_i, so the possibility of using Claim 2 for $n \geqslant 2^7 + 1$ is justified. It might seem that the choice of n is arbitrary, but it occurs that the distribution of M_{2^6+1} does not satisfy the necessary assumptions for privacy loss, although M_{2^6+1} can still fulfil the property of $(\varepsilon(n), \delta)$-DP with the parameters given in Theorem 1.

i	θ_i	2^{i-4}	$2^{4-i}\theta_i$
1	$9.6205\ldots \cdot 10^{-24}$	0.125	$7.6964\ldots \cdot 10^{-23}$
2	$1.73351\ldots \cdot 10^{-9}$	0.25	$6.93402\ldots \cdot 10^{-9}$
3	$0.000119359\ldots$	0.5	$0.000238718\ldots$
4	$0.0140238\ldots$	1	$0.0140238\ldots$
5	$0.158163\ldots$	2	$0.0790814\ldots$
6	$0.771817\ldots$	4	$0.192954\ldots$
7	$2.67702\ldots$	8	$0.334628\ldots$
8	$7.83367\ldots$	16	$0.489604\ldots$
9	$20.8095\ldots$	32	$0.650297\ldots$
10	$52.0472\ldots$	64	$0.813238\ldots$
11	$125.065\ldots$	128	$0.977073\ldots$

Tab. 1: Ratios of adjacent probabilities of the distribution of M_{2^7+1}, compared with the exponential function of base 2.

Lemma 3 *Let $k \in \mathbb{N} \setminus \{0\}$ and $n_k = 2^k + 1$. If $p_{n_k,k+c} \leqslant 2^{c+3} p_{n_k,k+c+1}$ for every c in the interval $[-k : 4]$, then*

$$(\forall\, N \geqslant n_k)(\forall\, c \in [-k:4])\ p_{N,k+c} < 2^{c+3} p_{N,k+c+1}\ .$$

Proof: Realise that for $c = -k$, the required inequality is trivial. Therefore, we can safely consider only $c \in [-k+1 : 4]$. We would like to prove this inductively with respect to c and N. Assume that for some $N \geqslant n_k$ and any $d \in \{c-1, c\}$ we have $p_{N,k+d} < 2^{d+3} p_{N,k+d+1}$.

Then also

$$\begin{aligned} p_{N+1,k+c} &\overset{(3)}{=} p_{N,k+c}(1 - 2^{-k-c}) + p_{N,k+c-1} 2^{-k-c+1} \\ &\leqslant 2^{3+c} p_{N,k+c+1}(1 - 2^{-k-c}) + 2^{3+(c-1)} p_{N,k+c} 2^{-k-c+1} \\ &< 2^{3+c}(p_{N,k+c+1}(1 - 2^{-k-c-1}) + 2^{-k-c} p_{N,k+c}) = 2^{3+c} p_{N+1,k+c+1}\ . \end{aligned}$$

If we start with $c = -k + 1$, then let us prove inductively the appropriate condition for all $N \geqslant n_k$. In addition, the thesis is followed by the induction with respect to c. □

ε **phase** Claims 1 and 2, together with Table 1 enable us to apply Lemma 3 for $n = 2^k + 1$ for any $k \geqslant 7$.

Theorem 6 *Let $n > 2^7 = 128$ and $k \in I_n$. Then*

$$1 - \frac{16}{n} \leqslant \frac{p_{n\pm1,k}}{p_{n,k}} \leqslant 1 + \frac{16}{n}\ .$$

Proof: According to the previous discussion on formula (10), we examine

$$\frac{p_{n+1,k}}{p_{n,k}} \overset{(3)}{=} \frac{p_{n,k}(1 - 2^{-k}) + 2^{-k+1} p_{n,k-1}}{p_{n,k}} = 1 + 2^{-k}\left(-1 + 2\frac{p_{n,k-1}}{p_{n,k}}\right)\ .$$

Let us denote $l = \lceil \log(n) \rceil$ and $c = k - l \in [-4:4]$. Then Lemma 3 bears $p_{n,k-1} \leqslant 2^{c+3} p_{n,k}$, so

$$\frac{p_{n+1,k}}{p_{n,k}} \leqslant 1 + 2^{-l-c}(-1 + 2^{c+4}) < 1 + 2^{-l+4} = 1 + \frac{16}{2^{\lceil \log(n) \rceil}} < 1 + \frac{16}{n} \ .$$

Realise that if $n = 2^{l-1} + 1$ for some $l \in \mathbb{N}$, then a little adjustment is necessary. Indeed, let now $c - 1 = k - l \in [-4:4]$, and once again, Lemma 3 provides $p_{n-1,k-1} < 2^{c+2} p_{n-1,k}$. However, it still holds that:

$$\frac{p_{n,k}}{p_{n-1,k}} = 1 + 2^{-k} \left(-1 + 2 \frac{p_{n-1,k-1}}{p_{n-1,k}} \right) \leqslant 1 + 2^{-l-c+1} \left(-1 + 2^{c+3} \right) < 1 + \frac{16}{n} \ .$$

On the other hand, we have inequalities $p_{n+1,k} > \left(1 - 2^{-l-c}\right) p_{n,k}$ and $p_{n,k} > \left(1 - 2^{-l-c}\right) p_{n-1,k}$ for any $c \in [-4:4]$, so both fractions exceed $1 - 16/n$. □

Theorem 6 only provides $\varepsilon(n) \leqslant -\ln(1 - 16/n)$ for $n > 128$ (compare with (10)). However, in Figure 5 we may briefly see that the above inequality is true for smaller numbers of requests n as well.

Having all the technical lemmas, we are now ready to prove Theorem 1.

Proof: (of Theorem 1) Suppose that $S_n = I_n$ in Fact 1. Then from theorems 5 and 6 we can easily see that $\mathbb{P}(M_n \notin S_n) < 0.00033$ and

$$(\forall\, m \in \{n-1, n+1\})\ (\forall\, l \in S_n)\ \mathbb{P}(M_n = l) \leqslant \left(1 - \frac{16}{n}\right)^{-1} \cdot \mathbb{P}(M_m = l) \ ,$$

hence, from Fact 1 we obtain the main result. □

In the Figure 5 one can see that values of $\varepsilon(n)$ are strictly between sequences $-\ln(1 - 8/n)$ and $-\ln(1 - 16/n)$ for $n \in [17:160]$. We can also observe that $\varepsilon(n) \approx 2^{4 - \lceil \log(n) \rceil}$ in this interval. Note that $\lceil \log(n) \rceil \leqslant 4$ for $n \leqslant 16$, so $\lceil \log(n) \rceil - 4 < 1$, but M is always positive. This can justify the chaotic behaviour of the process for $n \leqslant 16$. Nevertheless, Figure 5 affirms the quality of $\varepsilon(n)$ parameter established in Theorem 1.

Moreover, let us mention that for the strategy used in the proof, we cannot pick smaller ε of the same form:

Observation 2 *Assume that for I_n, defined in (4), $\delta = \Pr M_n \notin I_n$ and $\varepsilon(n)$, defined via (10), is of the form $|-\ln(1 - \frac{c}{n})|$. Then $c = 16$ is optimal constant in Theorem 1 (it cannot be improved). Indeed, see that the bound is reached for $n = 32$:*

$$\frac{p_{33,1}}{p_{32,1}} = \frac{1}{2} = 1 - \frac{16}{32} \ .$$

As can be seen in Figure 5, the minimal c for $\varepsilon(n)$, where n is of the form 2^k, for $k > 5$, is also very close to 16.

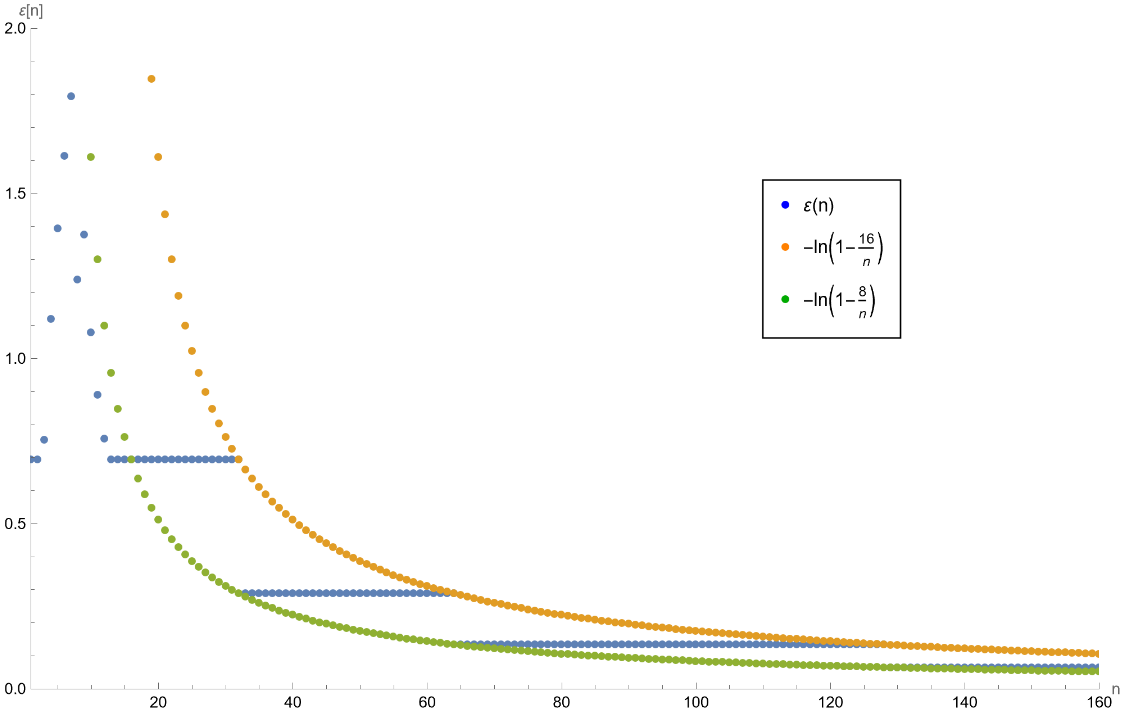

Fig. 5: Exact values of $\varepsilon(n)$ parameter for $n \leqslant 160$ compared with plots of sequences $-\ln(1-16/n)$ and $-\ln(1-8/n)$.

8.2 General result on Morris' Counter privacy (Theorem 2)

Remark that $\lceil c \log(\ln(n)) \rceil \geqslant 1$ can always be guaranteed, when n is large enough.

Proof: For our convenience, let us denote $\rho := \lceil c \log(\ln(n)) \rceil$. We assume that $\rho \geqslant 1$. First, we show that $\delta_1^* := \mathbb{P}\left(M_n \leqslant \lceil \log(n) \rceil - \rho - 1\right) = O\left(n^{-(\ln(n))^{c-1}}\right)$. The proof is analogous to the one of Lemma 1 (we omit similar parts). Indeed,

$$\delta_1^* = \sum_{l=1}^{\lceil \log(n) \rceil - \rho - 1} \mathbb{P}\left(M_n = l\right) \leqslant R^2(2\sqrt{2}+2) \sum_{k=\rho}^{\infty} \exp(-2^k)$$

$$\leqslant R^2(2\sqrt{2}+2) \sum_{k=1}^{\infty} \exp(-2^\rho k)$$

$$\leqslant R^2(2\sqrt{2}+2) \frac{\exp(-(\ln(n))^c)}{1-\exp(-(\ln(n))^c)} = O\left(n^{-(\ln(n))^{c-1}}\right) .$$

Now, we are going to prove $\delta_2^* := \mathbb{P}\left(M_n \geqslant \lceil \lg(n) \rceil + \rho + 1\right) = O\left(n^{-1}\left(\ln(n)\right)^{-c}\right)$. We use a similar notation and technique as in the proof of Lemma 2, but this time we utilise a discrete discrete two-dimensional simplex $(\rho - 1)$:

$$S_k^{(\rho-1)} = \left\{ \bar{l} = (l_1, l_2, \dots, l_{\rho-1}) \in \mathbb{N}_0^{\rho-1} : \sum_{i=1}^{\rho-1} l_i \leqslant k \right\} .$$

We couple $(M_n)_n$ with the same process $(X_n)_n$ as in Lemma 2. Roughly speaking, $X_0 = 1$ and X_n almost always increments by 1 until $n = \lceil \log(n) \rceil + 1$ and it further follows the same incrementation rule as the Morris counter. Then

$$\begin{aligned}
\delta_2^* &\leqslant \mathbb{P}\left(X_n \geqslant \lceil \log(n) \rceil + \rho + 1\right) = \sum_{\bar{l} \in S_{m-\rho+1}^{(\rho-1)}} \prod_{i=1}^{\rho-1} \nu_{i+1}^{l_i} \mu_i \leqslant \sum_{\bar{l} \in S_{m-\rho+1}^{(\rho-1)}} \prod_{i=1}^{\rho-1} \mu_i \\
&= \sum_{\bar{l} \in S_{m-\rho+1}^{(\rho-1)}} 2^{-[(\rho-1)\lceil \log(n) \rceil + \sum_{i=2}^{\rho} i]} \\
&= \sum_{k=0}^{m-\rho+1} \binom{k+\rho-1}{\rho-2} 2^{-[(\rho-1)\lceil \log(n) \rceil + \frac{(\rho+2)(\rho-1)}{2}]} \\
&\leqslant m^{\rho-2} n^{1-\rho} 2^{-\frac{\rho^2+\rho-2}{2}} \leqslant n^{-1} 2^{-\rho+1} = O\left(n^{-1} \left(\ln(n)\right)^{-c}\right) .
\end{aligned}$$

Therefore $\mathbb{P}\left(M_n \notin J_n(c)\right) = \delta_1^* + \delta_2^* = O\left(n^{-(\ln(n))^{c-1}} + n^{-1}\left(\ln(n)\right)^{-c}\right)$

In addition, we would like to consider fractions $\frac{p_{n+1,k}}{p_{n,k}}$ for $k \in J_n(c)$ as in the proof of Theorem 6. Indeed

$$1 - 2^{-k} \leqslant \frac{p_{n+1,k}}{p_{n,k}} = 1 - 2^{-k} + 2^{-k+1} \frac{p_{n,k-1}}{p_{n,k}} .$$

We are going to use another formula from Flajolet (1985). For any $n \in \mathbb{N}$ and $k \in [1 : n+1]$,

$$p_{n,k} = 2^{-\frac{k(k-1)}{2}} \sum_{\bar{l} \in S_{n-k+1}^{(k)}} \prod_{i=1}^{k} \left(1 - 2^{-i}\right)^{l_i} .$$

Let us denote the above sum by $\varsigma_k(n-k+1)$. We note that

$$2^{-k+1} \frac{p_{n,k-1}}{p_{n,k}} = 2^{-k+1} \frac{2^{-\frac{(k-2)(k-1)}{2}} \varsigma_{k-1}(n-k+2)}{2^{-\frac{k(k-1)}{2}} \varsigma_k(n-k+1)} = \frac{\varsigma_{k-1}(n-k+2)}{\varsigma_k(n-k+1)} .$$

Realise that $\varsigma_{k-1}(n-k+2) \leqslant \varsigma_{k-1}(n-k+1) \sum_{i=1}^{k-1} \left(1 - 2^{-i}\right) = \varsigma_{k-1}(n-k+1)\left(k - 2 + 2^{k-1}\right)$. This follows from the fact that each summand of $\varsigma_{k-1}(n-k+2)$ can be obtained from some summands of $\varsigma_{k-1}(n-k+1)$ by multiplication by one of the terms $\left(1 - 2^{-i}\right)$. Moreover, note that $\varsigma_k(n-k+1)$ has $\binom{n-k+1+(k-1)}{k-1} = \binom{n}{k-1}$ summands and, similarly, $\varsigma_{k-1}(n-k+1)$ has $\binom{n-1}{k-2}$ summands. One can briefly see that a function $f(i) = \left(1 - 2^{-i}\right)$ is increasing, hence

$$\begin{aligned}\varsigma_k(n-k+1) &\geqslant \sum_{\bar{l} \in S^{(k)}_{n-k+1}} \left(1 - 2^{-k+1}\right)^{l_{k-1}+l_k} \prod_{i=1}^{k-2} \left(1 - 2^{-i}\right)^{l_i} \\ &= \sum_{\bar{l} \in S^{(k-1)}_{n-k+1}} (l_{k-1}+1) \prod_{i=1}^{k-1} \left(1 - 2^{-i}\right)^{l_i} \ .\end{aligned}$$

Due to the monotonicity of f, one can use cascading substitutions: some of $f(k)$ by $f(k-1)$, then some of $f(k-1)$ by $f(k-2)$ etc., in order to balance the numbers of all the occurring summands, what provides:

$$\varsigma_k(n-k+1) \geqslant \sum_{\bar{l} \in S^{(k-1)}_{n-k+1}} \frac{\binom{n}{k-1}}{\binom{n-1}{k-2}} \prod_{i=1}^{k-1} \left(1 - 2^{-i}\right)^{l_i} = \frac{n}{k-1} \varsigma_{k-1}(n-k+1) \ .$$

Therefore $\varsigma_{k-1}(n-k+2) \leqslant \frac{\left(k-2+2^{k-1}\right)(k-1)}{n} \varsigma_k(n-k+1)$ and finally we obtain

$$\frac{p_{n+1,k}}{p_{n,k}} \leqslant 1 - 2^{-k} + \frac{\left(k - 2 + 2^{k-1}\right)(k-1)}{n} < 1 + \frac{(k-1)^2}{n} \ .$$

When $k \in J_n(c)$, then

$$\begin{aligned}\exp\left(-O\left(\frac{\log(n)}{n}\right)\right) = 1 - 2^{-\lceil \log(n) \rceil + \lceil \log \ln(n) \rceil} & \\ < \frac{p_{n+1,k}}{p_{n,k}} < & \\ 1 + \frac{\left(\lceil \log(n) \rceil + \log \ln(n)\right)^2}{n} &= \exp\left(O\left(\frac{(\log(n))^2}{n}\right)\right) \ .\end{aligned}$$

This shows that $\varepsilon(n) = O\left(\frac{(\log(n))^2}{n}\right)$. □

8.3 Main result for MaxGeo Counter (Theorem 3)

Proof: We have n increment requests, which influence the value of MaxGeo Counter M. Then the result of the mechanism is $X = \max(X_1, \ldots, X_n)$, where $X_i \sim \mathrm{Geo}(1/2)$ are pairwise independent. First, we observe that if $n = m$, the counter trivially satisfies differential privacy, as the probability distribution of X does not change. From now on, we assume that $|n - m| = 1$. See that

$$\mathbb{P}\left(X \leqslant l\right) = \prod_{i=1}^{n} \mathbb{P}\left(X_i \leqslant l\right) = \left(\mathbb{P}\left(X_1 \leqslant l\right)\right)^n = \left(1 - \frac{1}{2^l}\right)^n = \left(\frac{2^l - 1}{2^l}\right)^n \ .$$

Furthermore

$$\begin{aligned}\mathbb{P}\left(\max(X_1,\ldots,X_n)=l\right) &= \mathbb{P}\left(X=l\right)=\mathbb{P}\left(X\leqslant l\right)-\mathbb{P}\left(X\leqslant(l-1)\right)\\ &=\left(\frac{2^l-1}{2^l}\right)^n-\left(\frac{2^{l-1}-1}{2^{l-1}}\right)^n=\frac{\left(2^l-1\right)^n-\left(2^l-2\right)^n}{2^{l\cdot n}}\ .\end{aligned}$$

Now we need to calculate the following expression

$$\begin{aligned}\frac{\mathbb{P}\left(\max(X_1,\ldots X_n)=l\right)}{\mathbb{P}\left(\max(X_1,\ldots X_n,X_{n+1})=l\right)} &= \frac{\dfrac{\left(2^l-1\right)^n-\left(2^l-2\right)^n}{2^{l\cdot n}}}{\dfrac{\left(2^l-1\right)^{n+1}-\left(2^l-2\right)^{n+1}}{2^{l\cdot(n+1)}}}\\ &=\frac{\left(\left(2^l-1\right)^n-(2^l-2)^n\right)\cdot 2^l}{\left(2^l-1\right)^{n+1}-(2^l-2)^{n+1}}\\ &=\frac{2^l}{2^l-1}\cdot\frac{\left(\left(2^l-1\right)^n-(2^l-2)^n\right)}{\left(\left(2^l-1\right)^n-\frac{(2^l-2)^{n+1}}{2^l-1}\right)}\\ &\leqslant\frac{2^l}{2^l-1}\cdot\frac{\left(\left(2^l-1\right)^n-(2^l-2)^n\right)}{\left(\left(2^l-1\right)^n-\frac{(2^l-2)^{n+1}}{2^l-2}\right)}\\ &=\frac{2^l}{2^l-1}=1+\frac{1}{2^l-1}\ .\end{aligned}$$

For fixed ε we need to satisfy the following inequality

$$\left|\ln\left(\frac{P(\max(X_1,\ldots X_n)=l)}{P(\max(X_1,\ldots X_n,X_{n+1})=l)}\right)\right|\leqslant\varepsilon\ ,$$

which gives

$$\ln\left(1+\frac{1}{2^l-1}\right)\leqslant\varepsilon\ . \tag{11}$$

We can see from (11) that the greater l is, the smaller ε can be. Moreover, inequality (11) is true for $l\geqslant l_\varepsilon$. Therefore, we must ensure $P(X\leqslant l_\varepsilon)\leqslant\delta$. See that

$$\mathbb{P}\left(X\leqslant l_\varepsilon\right)=\left(1-2^{-l_\varepsilon}\right)^n\ .$$

It is easy to see that the above decreases as n increases. Then

$$\left(1-2^{-l_\varepsilon}\right)^n\leqslant\delta\iff n\geqslant\frac{\ln(\delta)}{\ln(1-2^{-l_\varepsilon})}\approx-\frac{\ln(\delta)}{\varepsilon},$$

where the approximation is the result of the substitution of l_ε without ceiling. □

9 Conclusions and Future Work

In this paper, we have investigated probabilistic counters from a privacy-protection perspective. We have shown that Morris Counter and MaxGeo Counter inherently guarantee differential privacy from the mechanism itself, provided that there is at least a small fixed number of increment requests. Otherwise, the counter has too low a value and, intuitively, the result is not randomised enough. We have also shown that the constant in our Morris Counter result cannot be improved further.

We have shown how to perform data aggregation, namely a distributed survey, in a privacy-preserving manner using probabilistic counters. We clarified that this type of solution is especially efficient when one cares about memory resources, like in many Big Data related problems. Note that the security model in this paper was somewhat optimistic. Unfortunately, in such a setting, there is little incentive to use them other than when we already have them deployed and working as aggregators due to e.g., memory-efficiency requirements. However, this would change tremendously if we weakened these assumptions. This seems a very promising way to continue our research from this paper. Namely, we focused on privacy and can still not weaken the security assumptions and allow the Adversary to extract information from channels between users and the aggregator. That would put us in the so-called Local Model, where each user is responsible for the data randomisation. However, such an approach requires us to be able to perform probabilistic counter in an oblivious manner, which, to the best of our knowledge, was not explored before.

In Subsection 4.1, we have mentioned the generalisation of the Morris Counter (for bases $a > 1$). Analysis of privacy properties of such variants of Morris Counters and various probabilistic counters presented, for example, in Csűrös (2010), Fuchs et al. (2012) may also be promising directions of further research.

In this paper, we focus on the standard definition of differential privacy. However, there is also an issue of preservation of differential privacy for requests given by a group of k users or one individual sending up to k dependent requests over time. This can be described in the language of the so-called $(\varepsilon, \delta) - k$-DP (see the "group privacy scenario" in Dwork and Roth (2014)) . A group of people may tend to behave in the same manner, so they may send k requests in a row. Especially this "group" may be represented by a single person colluding with the Adversary. It is worth mentioning that this type of generalisation creates an opportunity to modify probabilistic counters so that each incrementation request executes the update request multiple times to reduce the variance of the rescaled estimator. Intuitively, this extension should be especially efficient in preserving the standard differential privacy property when $\varepsilon(n) = \frac{c}{n} + o(n^{-1})$ (as a parameter of standard differential privacy), because both c and n should scale with k linearly. Hence, the next challenging problem is to show that Morris and MaxGeo Counters satisfy the k-DP property with similar privacy parameters.

The Morris Counter and the MaxGeo Counter are considered the most popular probabilistic counters. However, the results of this paper shed new light on the properties of the probabilistic counter, in general. There is a possibility to provide analogous differential privacy properties for other probabilistic counters. Moreover, this paper enables the provision of differentially private algorithms for other applications, especially those based on Morris or MaxGeo Counter. For example, in Section 4 we mentioned PCSA and HyperLogLog Counters together with their variances, which can be manually adjusted to the applications. The proper choice of the m parameter implies an exchange of memory usage to improve the accuracy of the estimation. We have mentioned that these counters' differential privacy parameters can be obtained via Observation 1. However, such a direct result may not be satisfying. Hence a more precise calculation is needed. For example, Observation 1 may be used again with some concentration inequalities.

Acknowledgements

The authors sincerely thank the reviewers and editors for their significant time and effort invested in editing and improving this work. This work was supported by the Polish National Science Center grant number UMO-2018/29/B/ST6/02969.

References

N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137 – 147, 1999.

C. Baquero, P. Almeida, and R. Menezes. Fast estimation of aggregates in unstructured networks. In *Twenty-First International Joint Conference on Artificial Intelligence*, pages 88–93, 2009.

B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.

B. Bollobás. Random graphs. In *Modern Graph Theory*, pages 215–252. Springer, 1998.

S. G. Choi, D. Dachman-Soled, M. Kulkarni, and A. Yerukhimovich. Differentially-private multi-party sketching for large-scale statistics. *Proc. Priv. Enhancing Technol.*, 2020(3):153–174, 2020.

J. Cichoń and K. Gotfryd. Average counting via approximate histograms. *ACM Trans. Sen. Netw.*, 14(2), 2018.

J. Cichon and W. Macyna. Approximate counters for flash memory. *2011 IEEE 17th International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 185–189, 2011.

E. Cohen. All-distances sketches, revisited. *IEEE TKDE*, 2015.

D. Crippa and K. Simon. Q-distributions and markov processes. *Discrete Math.*, 170(1):81–98, 1997.

M. Csűrös. Approximate counting with a floating-point counter. pages 358–367, 2010.

D. Desfontaines, A. Lochbihler, and D. Basin. Cardinality estimators do not preserve privacy. *Proceedings on Privacy Enhancing Technologies*, 2019(2):26–46, 2019.

D. Dice, Y. Lev, and M. Moir. Scalable statistics counters. In *Proceedings of the twenty-fifth annual ACM symposium on Parallelism in algorithms and architectures*, pages 43–52, 2013.

M. Durand and P. Flajolet. Loglog counting of large cardinalities. In *European Symposium on Algorithms*, pages 605–617. Springer, 2003.

C. Dwork. Differential privacy. In *ICALP 2006*, pages 1–12, 2006.

C. Dwork and J. Lei. Differential privacy and robust statistics. In *STOC 2009*, pages 371–380, 2009.

C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9:211–407, 2014.

C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503, 2006a.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC 2006*, pages 265–284, 2006b.

C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724, 2010.

G. Einziger, B. Fellman, R. Friedman, and Y. Kassner. Ice buckets: Improved counter estimation for network measurement. *IEEE/ACM Transactions on Networking*, 2018.

B. Eisenberg. On the expectation of the maximum of iid geometric random variables. *Statistics & Probability Letters*, 78(2):135–143, 2008.

P. Flajolet. Approximate counting: a detailed analysis. *BIT Numerical Mathematics*, 25, 1985.

P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *Journal of computer and system sciences*, 31:182–209, 1985.

P. Flajolet, É. Fusy, O. Gandouet, and F. Meunier. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. In *AofA 2007*, 2007.

A. Frieze and M. Karonski. *Introduction to Random Graphs*. Cambridge University Press, 2015.

M. Fuchs, C.-K. Lee, and H. Prodinger. Approximate Counting via the Poisson-Laplace-Mellin Method. In *23rd International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms (AofA'12)*, pages 13–28. Discrete Mathematics and Theoretical Computer Science, 2012.

A. Gronemeier and M. Sauerhoff. Applying approximate counting for computing the frequency moments of long data streams. *Theory of Computing Systems*, 2009.

S. Heule, M. Nunkesser, and A. Hall. Hyperloglog in practice: algorithmic engineering of a state of the art cardinality estimation algorithm. In *EDBT 2013*, 2013.

P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000*, pages 189–197. IEEE Computer Society, 2000. doi: 10.1109/SFCS.2000.892082. URL `https://doi.org/10.1109/SFCS.2000.892082`.

P. Indyk and D. Woodruff. Tight lower bounds for the distinct elements problem. In *FOCS 2003*, 2003.

M. Klonowski and A. M. Piotrowska. Light-weight and secure aggregation protocols based on bloom filters. *Comput. Secur.*, 72:107–121, 2018. doi: 10.1016/j.cose.2017.08.015.

K.-H. Li. Reservoir-sampling algorithms of time complexity $o(n(1 + log(n/n)))$. *ACM Trans. Math. Softw.*, 20(4):481–493, 1994.

H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang. Learning differentially private recurrent language models. 2018. arXiv:1710.06963.

N. Mishra and M. Sandler. Privacy via pseudorandom sketches. In *Proceedings of the Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 143–152. ACM, 2006. doi: 10.1145/1142351.1142373.

R. Morris. Counting large numbers of events in small registers. *Communications of the ACM*, 1978.

A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *30th IEEE Symposium on Security and Privacy*, 2009.

A. Narayanan and V. Shmatikov. Myths and fallacies of personally identifiable information. *Communications of the ACM*, 53(6):24–26, 2010.

J. Nelson and H. Yu. Optimal bounds for approximate counting, 2022. arXiv:2010.02116.

J. Neyman. Outline of a theory of statistical estimation based on the classical theory of probability. *Philosophical Transactions of the Royal Society of London. Series A*, 236(767):333–380, 1937.

C. R. Palmer, P. B. Gibbons, and C. Faloutsos. Anf: A fast and scalable tool for data mining in massive graphs. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data. 2017. arXiv:1610.05755.

N. Riley and C. Zilles. Probabilistic counter updates for predictor hysteresis and bias. *IEEE Computer Architecture Letters*, 5(1):18–21, 2006.

R. Sedgewick. Cardinality estimation, 2018. URL `https://www.cs.princeton.edu/~rs/talks/CardinalityX.pdf`.

A. D. Smith, S. Song, and A. Thakurta. The flajolet-martin sketch itself preserves differential privacy: Private counting with minimal space. *Advances in Neural Information Processing Systems 33*, 2020.

S. J. Swamidass and P. Baldi. Mathematical correction for fingerprint similarity measures to improve chemical retrieval. *J. Chem. Inf. Model.*, 47(3):952–964, 2007. doi: 10.1021/ci600526a.

W. Szpankowski and V. Rego. Yet another application of a binomial recurrence order statistics. *Computing*, 1990.

D. Ting. Streamed approximate counting of distinct elements. In *KDD '14*, 2014.

D. Ting. Approximate distinct counts for billions of datasets. In *SIGMOD '19*, 2019.

B. Van Durme and A. Lall. Probabilistic counting with randomized storage. In *IJCAI 2009*, 2009.

T. Wang, J. Zhao, H. Yu, J. Liu, X. Yang, X. Ren, and S. Shi. Privacy-preserving crowd-guided ai decision-making. In *CIKM '19*, 2019.

Wolfram Research, Inc. Mathematica, Version 14.1, 2025. URL `https://www.wolfram.com`.

X. Yun, G. Wu, G. Zhang, K. Li, and S. Wang. Fastraq: A fast approach to range-aggregate queries in big data environments. *IEEE Transactions on Cloud Computing*, 3(2):206–218, 2015. doi: 10.1109/TCC.2014.2338325.

A Technical Lemmas and Proofs Related to Differential Privacy of Morris Counter

For the sake of completeness, we present here proofs of all technical lemmas that are not directly connected to Theorem 1. Some of computations are supported by Wolfram Mathematica ver.11.3 (Wolfram Research, Inc.). Whenever we obtain a result in this manner, we indicate it by $\overset{W}{=}$ sign. Usually results are precise, however in some cases, final forms are attained numerically.

We often struggle with expressions of a pattern $1 - 1/y$, so let us denote this function as $a(y)$ to abbreviate formulas.

Next two lemmas will be useful in a proof of Lemma 6.

Lemma 4 *Let* $c > 1/x$. *Then*

$$a(2cx)^{2y} \geqslant a(cx)^{y-1}\left(a(cx) + \frac{y}{4c^2x^2}\right)$$

and

$$a(cx)^{y} \geqslant a(2cx)^{2y-2}\left(a(2cx)^2 - \frac{y}{4c^2x^2}\right) .$$

Proof:

$$\frac{a(2cx)^{2y} - a(cx)^y}{a(2cx)^2 - a(cx)} = \sum_{i=0}^{y-1} a(2cx)^{2i} a(cx)^{y-i-1} .$$

Realize that the above denominator is $\left(1 - \frac{1}{cx} + \frac{1}{4c^2x^2} - 1 + \frac{1}{cx}\right) = \frac{1}{4c^2x^2}$. Hence, we obtained two inequalities: $a(2cx)^{2y} - a(cx)^y \geqslant \dfrac{y}{4c^2x^2} a(cx)^{y-1}$
and $a(2cx)^{2y} - a(cx)^y \leqslant \dfrac{y}{4c^2x^2}\, a(2cx)^{2(y-1)}$, which imply the thesis of this Lemma. □

Lemma 5 *Let* $s \leqslant \log(x/4)$. *Then* $a(2^{-s}x)^{2x+1} < \exp(-2^{s+1})$ *and*

$$a(2^{-s}x)^{x-1} > \exp(-2^s)\left(1 - \frac{2^{2s-1} - 2^s}{x} - \frac{2^{2s-7} + 2^{4s-3}}{x^2}\right) .$$

Proof: Let $f_1(x;s) := a(2^{-s}x)^{2x+1}$. For any $s \leqslant \log(x/4)$, we have $f_1(x;s) = \left(1 - \frac{2^{2+1}}{2x}\right)^{2x=1} = \exp(-2^{s+1})\left(1 - O\left(x^{-1}\right)\right)$. Realize a fact, that $z\ln(z) \geqslant z - 1$, for $0 < z \leqslant 1$. Hence

$$\left(1-\frac{2^s}{x}\right)^{-2x}\frac{\partial f_1(x;s)}{\partial x}=\frac{2^s(2x+1)}{x^2}+2\left(1-\frac{2^s}{x}\right)\ln\left(1-\frac{2^s}{x}\right)>\frac{2^s}{x^2}>0$$

and in a consequence $a(2^{-s}x)^{2x+1}<\exp(-2^{s+1})$ for any reasonable s.

Moreover, let us introduce

$$D(x;s):=1-\frac{2^{2s-1}-2^s}{x}-\frac{2^{2s-7}+2^{4s-3}}{x^2}\,.$$

It is defined in such the way that $a(2^{-s}x)^{x-1}=D(x;s)+O(x^{-2})$. Therefore we can attain:

$$f_2(x;s):=\frac{a(2^{-s}x)^{x-1}}{D(x;s)}=\exp(-2^s)\left(1+O\left(x^{-2}\right)\right)\,.$$

Then, in a similar way

$$\begin{aligned}
&D(x;s)^2\left(1-\frac{2^s}{x}\right)^{-x+1}\frac{\partial f_2(x;s)}{\partial x}=\\
&D(x;s)\left(\frac{2^s(x-1)}{x^2\left(1-\frac{2^s}{x}\right)}+\ln\left(1-\frac{2^s}{x}\right)\right)-\left(\frac{2^{2s-6}+2^{4s-2}}{x^3}+\frac{2^{2s-1}-2^s}{x^2}\right)\\
&<\left(\frac{2^{2s-1}-2^s}{x^2}+\frac{2^{3s}-2^{2s}}{x^3\left(1-\frac{2^s}{x}\right)}-\frac{2^{3s}}{3x^3}\right)-\left(\frac{2^{2s-6}+2^{4s-2}}{x^3}+\frac{2^{2s-1}-2^s}{x^2}\right)\\
&=\frac{2^{3s}-2^{2s}}{x^3\left(1-\frac{2^s}{x}\right)}-\frac{2^{3s}}{3x^3}-\frac{2^{2s-6}+2^{4s-2}}{x^3}\,.
\end{aligned}$$

Let $d:=1-2^s/x$ and realize that $d\in[3/4,1)$ and $2^s-1+d\left(-2^s/3-2^{-6}-2^{2s-2}\right)>0$. Indeed, if we put $z=2^s$, then we attain a quadratic inequality in z variable, with determinant $\Delta=1-\frac{5d}{3}+\frac{55d^2}{576}$, that is negative for $d\in[3/4,1)$.

Hence $\frac{\partial f_2(x;s)}{\partial x}<0$ and consequently

$$a(2^{-s}x)^{x-1}>\exp(-2^s)\left(1-\frac{2^{2s-1}-2^s}{x}-\frac{2^{2s-7}+2^{4s-3}}{x^2}\right)$$

for any reasonable s. □

Lemma 6 *a) The sequence $(p_{2^k+1,k+4})_{k=2}^{\infty}$ is descending.*

b) The sequence $(p_{2^k+1,k+5})_{k=3}^{\infty}$ is ascending.

Proof: Let $x=2^k$ and $t\in\{0,1\}$. In advance we define

$$\kappa(k,t):=(-1)^{k+4+t}2^{-\frac{(k+4+t)(k+3+t)}{2}}r_{k+4+t}2^{-2x-1}$$

and

$$\tau(k,t) := [\![2 \nmid (k+t)]\!](-1)^{k+t+3}2^{-\frac{(k+t+3)(k+t+2)}{2}}2r_{k+t+3}\left(\left(\frac{3}{4}\right)^{2x+1}-\left(\frac{1}{2}\right)^{x+2}\right) ,$$

where $[\![\mathrm{cond}]\!]$ is the Iverson bracket of the condition cond.

Realize that for $t \in \{0,1\}$ and $k \geqslant 5$, $|\tau(k,t)+\kappa(k,t)| < 2^{-50} < 10^{-15}$. Now, consider the differences between the consecutive elements of sequences:

$$\begin{aligned}
&p_{2^{k+1}+1,k+5+t} - p_{2^k+1,k+4+t} \overset{\text{Thm 4}}{=} \kappa(k,t)+\\
&+\sum_{i=0}^{k+3+t}(-1)^i 2^{-\frac{i(i-1)}{2}} r_i r_{k+t+4-i}\left[\left(1-\frac{2^{-5-t+i}}{x}\right)^{2x+1}-\left(1-\frac{2^{-4-t+i}}{x}\right)^{x+2}\right]\\
&=\sum_{i=0}^{\lfloor\frac{k+2+t}{2}\rfloor}\Big\{2^{-i(2i-1)}r_{2i}r_{k+t+4-2i}\left[a(2^{5+t-2i}x)^{2x+1}-a(2^{4+t-2i}x)^{x+2}\right]\\
&-2^{-(2i+1)i}r_{2i+1}r_{k+t+3-2i}\left[a(2^{4+t-2i}x)^{2x+1}-a(2^{3+t-2i}x)^{x+2}\right]\Big\}+(\tau+\kappa)(k,t)\\
&=\sum_{i=0}^{\lfloor\frac{k+t+2}{2}\rfloor} 2^{-i(2i-1)}r_{2i+1}r_{k+t+4-2i}\Big[a(2^{2i+1})\left(a(2^{5+t-2i}x)^{2x+1}-a(2^{4+t-2i}x)^{x+2}\right)\\
&-2^{-2i}a(2^{4+t-2i}x)\left(a(2^{4+t-2i}x)^{2x+1}-a(2^{3+t-2i}x)^{x+2}\right)\Big]+\tau(k,t)+\kappa(k,t) .
\end{aligned}$$

Let us define $u_t := 2^{5+t-2i}$ (note that u_t depends on i, but we abbreviate the notation for conciseness) and

$$\begin{aligned}
W_t(i) := a(2^{2i+1})&\left(a(u_t x)^{2x+1}-a\left(\frac{u_t}{2}x\right)^{x+2}\right)\\
-2^{-2i}a\left(\frac{u_t}{2}x\right)&\left(a\left(\frac{u_t}{2}x\right)^{2x+1}-a\left(\frac{u_t}{4}x\right)^{x+2}\right)
\end{aligned}$$

and consider an upper bound of the last term:

$$\begin{aligned}
W_t(i) &\overset{\text{Lem.4}}{\leqslant} a(2^{2i+1})\left(a(u_t x)^{2x+1}-a(u_t x)^{2x+2}\left(a(u_t x)^2-\frac{x+2}{u_t^2x^2}\right)\right)\\
&-2^{-2i}\left(a\left(\frac{u_t}{4}x\right)^x\left(a\left(\frac{u_t}{4}x\right)+\frac{x+1}{\frac{u_t^2}{4}x^2}\right)-a\left(\frac{u_t}{2}x\right)a\left(\frac{u_t}{4}x\right)^{x+2}\right)\\
&\overset{W}{=} a(2^{2i+1})a(u_t x)^{2x+1}\frac{1}{x}\left(\frac{3u_t+1}{u_t^2}-\frac{u_t+1}{u_t^3x}-\frac{1}{u_t^3x^2}\right)\\
&-2^{-2i}a\left(\frac{u_t}{4}x\right)^x\frac{1}{x}\left(\frac{6u_t+4}{u_t^2}-\frac{28}{xu_t^2}+\frac{32}{x^2u_t^3}\right) .
\end{aligned}$$

Note that $2i \leqslant k+2+t$, so $8/x \leqslant u_t$ and in consequence:

$6u_t - 28/x > 20/x > 0,$

$$\begin{aligned} u_t \cdot (3u_t+1) - \frac{u_t+1}{x} - \frac{1}{x^2} &\geqslant u_t \cdot \left(\frac{24}{x}+1\right) - \frac{u_t+1}{x} - \frac{1}{x^2} \\ &\geqslant \frac{8}{x} - \frac{1}{x} + \frac{23u_t}{x} - \frac{1}{x^2} \geqslant \frac{7}{x} + \frac{183}{x^2} > 0\,. \end{aligned}$$

Hence

$$\begin{aligned} &W_t(i) \overset{\text{Lem.5}}{<} a(2^{2i+1}) \exp\left(-\frac{2}{u_t}\right) \frac{1}{x} \left(\frac{3u_t+1}{u_t^2} - \frac{u_t+1}{xu_t^3} - \frac{1}{x^2u_t^3}\right) \\ &- 2^{-2i} a\left(\frac{u_t}{4}x\right) \exp\left(-\frac{4}{u_t}\right) D(x; 2i-3-t) \frac{1}{x} \left(\frac{6u_t+4}{u_t^2} - \frac{28}{xu_t^2} + \frac{32}{x^2u_t^3}\right) \\ &\overset{W}{=} a(2^{2i+1}) \exp\left(-\frac{2}{u_t}\right) \frac{1}{x} \left(\frac{3u_t+1}{u_t^2} - \frac{u_t+1}{xu_t^3} - \frac{1}{x^2u_t^3}\right) \qquad (12) \\ &- \frac{2^{-2i}}{x} \exp\left(-\frac{4}{u_t}\right) \Bigg(\frac{6u_t+4}{u_t^2} - \frac{32+48u_t+28u_t^2}{u_t^4x} - \frac{128+64u_t-\frac{703}{2}u_t^2+\frac{259}{4}u_t^3}{u_t^6x^2} \\ &+ \frac{512+128u_t-1150u_t^2+\frac{909}{2}u_t^3}{u_t^7x^3} - \frac{4608-1024u_t+530u_t^2}{u_t^7x^4} + \frac{4096+16u_t^2}{u_t^8x^5}\Bigg) \end{aligned}$$

We denote the upper bound (12) by $U_t(x; u_t(i))$.

Analogically we would like to establish a lower bound of $W_t(i)$:

$$\begin{aligned} W_t(i) \overset{\text{Lem.4}}{\geqslant} &a(2^{2i+1}) \left(a(u_tx) a\left(\frac{u_t}{2}x\right)^{x-1} \left(a\left(\frac{u_t}{2}x\right) + \frac{1}{u_t^2x}\right) - a\left(\frac{u_t}{2}x\right)^{x+2}\right) \\ &-2^{-2i} a\left(\frac{u_t}{2}x\right) \left(a\left(\frac{u_t}{2}x\right)^{2x+1} - a\left(\frac{u_t}{2}x\right)^{2x+2} \left(a\left(\frac{u_t}{2}x\right)^2 - \frac{x+2}{\frac{u_t^2}{4}x^2}\right)\right) \\ \overset{W}{=} &a(2^{2i+1}) a\left(\frac{u_t}{2}x\right)^{x-1} \frac{1}{x} \left(\frac{3u_t+1}{u_t^2} - \frac{10u_t+1}{u_t^3x} + \frac{8}{u_t^3x^2}\right) \qquad (13) \\ &-2^{-2i} a\left(\frac{u_t}{2}x\right)^{2x+2} \frac{1}{x} \left(\frac{6u_t+4}{u_t^2} - \frac{4u_t+8}{u_t^3x} - \frac{8}{u_t^3x^2}\right) \end{aligned}$$

Now from $8/x \leqslant u_t$ we attain

$$\begin{aligned} u_t \cdot (3u_t+1) - \frac{10u_t+1}{x} &= u_t \cdot \left(\frac{7u_t}{4} + \frac{7}{8}\right) + (10u_t+1)\left(\frac{u_t}{8} - \frac{1}{x}\right) \\ &\geqslant u_t \cdot \left(\frac{7u_t}{4} + \frac{7}{8}\right) > 0 \qquad (14) \end{aligned}$$

and

$$u_t \cdot (6u_t+4) - \frac{4u_t+8}{x} - \frac{8}{x^2} \geqslant \frac{48u_t}{x} + \frac{32}{x} - \frac{4u_t+8}{x} - \frac{8}{x^2} \geqslant \frac{24}{x} + \frac{344}{x^2} > 0\,.$$

Hence

$$\begin{aligned}W_t(i) &\overset{\text{Lem.5}}{>} a(2^{2i+1})\exp\left(-\frac{2}{u_t}\right)\frac{D(x;2i-4-t)}{x}\left(\frac{3u_t+1}{u_t^2}-\frac{10u_t+1}{u_t^3x}+\frac{8}{u_t^3x^2}\right)\\ &-2^{-2i}\exp\left(-\frac{4}{u_t}\right)a\left(\frac{u_t}{2}x\right)\frac{1}{x}\left(\frac{6u_t+4}{u_t^2}-\frac{4u_t+8}{u_t^3x}-\frac{8}{u_t^3x^2}\right)\\ &\overset{W}{=}\frac{a(2^{2i+1})}{x}\exp\left(-\frac{2}{u_t}\right)\left(\frac{3u_t+1}{u_t^2}-\frac{2+5u_t+4u_t^2}{u_t^4x}-\frac{2+4u_t-\frac{575}{32}u_t^2+\frac{387}{32}u_t^3}{u_t^6x^2}\right.\\ &\left.+\frac{2+20u_t-\frac{511}{32}u_t^2+\frac{261}{16}u_t^3}{u_t^7x^3}-\frac{16+\frac{1}{4}u_t^2}{u_t^7x^4}\right)\\ &-2^{-2i}\exp\left(-\frac{4}{u_t}\right)\frac{1}{x}\left(\frac{6u_t+4}{u_t^2}-\frac{16u_t+16}{u_t^3x}+\frac{16}{u_t^4x^2}+\frac{16}{u_t^4x^3}\right)\ .\end{aligned} \tag{15}$$

Denote the lower bound (15) by $L_t(x;u_t(i))$.

Now we show that $W_t(i) > 0$ for $i \geqslant 1$. Indeed, after reducing the redundant terms from inequality (13), together with inequality (14), we can obtain

$$W_t(i) > \frac{a\left(\frac{u_t}{2}x\right)^x}{xu_t^2}\left(a(2^{2i+1})\frac{14u_t+7}{8}-2^{-2i}(6u_t+4)\right) \tag{16}$$

Let us denote $C(t,x) := \frac{a\left(\frac{u_t}{2}x\right)^x}{xu_t^2}$. In case $i \geqslant 2$, we may attain

$$(16) \geqslant \frac{C(t,x)}{256}\left(31(14u_t+7)-96u_t-64\right) > 0.$$

When $i = 1$, then $u_t \geqslant 8$, so $(16) \geqslant \frac{C(t,x)}{64}\left(7(14u_t+7)-96u_t-64\right) = \frac{2u_t-15}{64} \geqslant \frac{1}{64}$.

Thanks to the property $W_t(i) > 0$ for $i \geqslant 1$, we may subtly neutralize the influence of r_{k+5-i} in the considered sum:

$$\sum_{i=0}^{\left\lfloor\frac{k+2}{2}\right\rfloor} 2^{-i(2i-1)}r_{2i+1}r_{k+5-i}W_0(i) < r_{k+5}\sum_{i=0}^{\left\lfloor\frac{k+2}{2}\right\rfloor} 2^{-i(2i-1)}r_{2i+1}W_0(i)\ .$$

Naturally we may consider $U_0(x;u_0(i))$ instead of $W_0(i)$ numerically for $i \leqslant 4$:

$$\begin{aligned}\sum_{i=0}^{4} 2^{-i(2i-1)}r_{2i+1}U_0(x;u_0(i)) &\overset{W}{=} -8.294491525704523\ldots\cdot 10^{-6}+\frac{0.15588\ldots}{x}\\ &+\frac{0.00407163\ldots}{x^2}-\frac{0.0298032\ldots}{x^3}+\frac{0.0198815\ldots}{x^4}-\frac{0.00785419\ldots}{x^5}\ ,\end{aligned}$$

so for $x \geqslant 2^{15}$ $(k \geqslant 15)$, $\sum_{i=0}^{4} 2^{-i(2i-1)}r_{2i+1}W_0(i) \leqslant -3.53741\cdot 10^{-6}$. Moreover we may bound $W_0(i)$ by $a(2^{5-2i}x)^{2x+1}$ for the rest of the sum:

$$\sum_{i=5}^{\left\lfloor\frac{k+2}{2}\right\rfloor} 2^{-i(2i-1)}r_{2i+1}a(2^{5-2i}x)^{2x+1} \leqslant \frac{R\,2^{-45}\exp(-64)}{1-2^{-21}\exp(-192)} = 1.5784\ldots\cdot 10^{-41}\ ,$$

so $p_{2^{k+1}+1,k+5} - p_{2^k+1,k+4} < 0$ for $k \geqslant 15$.
However, according to Theorem 4, we also present the numerical values of the sequence $(p_{2^k+1,k+4})_{k=2}^{14}$ in the Table 2. We can now easily see that for any $k \geqslant 2$ we attained $p_{2^{k+1}+1,k+5} - p_{2^k+1,k+4} < 0$.

k	$p_{2^k+1,k+4}$	k	$p_{2^k+1,k+4}$	k	$p_{2^k+1,k+4}$
2	0.0000305176 . . .	7	0.0000189841 . . .	12	0.0000185484 . . .
3	0.0000256707 . . .	8	0.0000187590 . . .	13	0.0000185413 . . .
4	0.0000221583 . . .	9	0.0000186466 . . .	14	0.0000185378 . . .
5	0.0000203424 . . .	10	0.0000185904 . . .		
6	0.0000194356 . . .	11	0.0000185624 . . .		

Tab. 2: Numerical values of the sequence $(p_{2^k+1,k+4})_{k=2}^{14}$.

Moreover, realize that $r_{k+5}/r_{k+3} < 1.1$ for any $k \geqslant 3$, so

$$\sum_{i=0}^{1} 2^{-i(2i-1)} r_{2i+1} 1.1^{1-i} L_1(x; u_1(i)) \overset{W}{=} 0.00128843\ldots + \frac{0.00212699\ldots}{x}$$
$$-\frac{0.00326251\ldots}{x^2} + \frac{0.000219133}{x^3} - \frac{3.50924875\ldots \cdot 10^{-7}}{x^4}$$

For any possible $x \geqslant 8$ ($k \geqslant 3$), $\sum_{i=0}^{1} 2^{-i(2i-1)} r_{2i+1} 1.1^{1-i} L_1(x; u_t(i)) > 0.0015$. We already know that $W_1(i)$ are positive for $i > 1$, so $p_{2^{k+1}+1,k+6} - p_{2^k+1,k+5} > 0$ for all $k \geqslant 3$. □

We may use Theorem 4 once again to see that

$$\frac{p_{2^6+1,10}}{p_{2^6+1,11}} = 129.454\ldots > 2^7 \text{and,}$$

$$\frac{p_{2^7+1,11}}{p_{2^7+1,12}} = 125.065\ldots < 2^7.$$

Together with Lemma 6 we may easily attain Claim 1 and we instantly see that this Claim cannot be extended continuously for $k < 7$.

Lemma 7 *Let* $2 \leqslant l \leqslant n$ *and assume that* $\alpha_i = 2^{i-2} \frac{p_{n,l-i}}{p_{n,l-i+1}}$ *for* $i \in [0:2]$ *and* $\alpha'_j = 2^{j-2} \frac{p_{n+1,l-j}}{p_{n+1,l-j+1}}$ *for* $j \in [0:1]$.
If $0 \leqslant \alpha_2 < \alpha_1 < \alpha_0$, *then* $0 < \alpha'_1 < \alpha'_0$.

Proof: Realize that $p_{n+1,l-i+1} = p_{n,l-i+1}(1 - 2^{-l+i-1} + 2^{-l+2}\alpha_i)$ for $i \in [0:2]$, so for $j \in [0:1]$,

$$\alpha'_j = \frac{p_{n+1,l-j}}{2^{2-j} p_{n+1,l-j+1}} = \frac{p_{n,l-j}(1 - 2^{-l+j} + 2^{-l+2}\alpha_{j+1})}{2^{2-j} p_{n,l-j+1}(1 - 2^{-l+j-1} + 2^{-l+2}\alpha_j)}$$
$$= \frac{\alpha_j(1 - 2^{-l+j} + 2^{-l+2}\alpha_{j+1})}{1 - 2^{-l+j-1} + 2^{-l+2}\alpha_j} .$$

Assume that $\alpha_1' \geqslant \alpha_0'$. Then

$$A := \alpha_1(1 - 2^{-l+1} + 2^{-l+2}\alpha_2)(1 - 2^{-l-1} + 2^{-l+2}\alpha_0) \geqslant \alpha_0(1 - 2^{-l} + 2^{-l+2}\alpha_1)^2 =: B.$$

However, contrary to the assumption,

$$\begin{aligned} A = {} & \alpha_1(1 - 2^{-l+1} + 2^{-2l} - 2^{-l-1} + 2^{-l+2}(\alpha_0 + \alpha_2) - 2^{-2l+3}\alpha_0 \\ & - 2^{-2l+1}\alpha_2 + 2^{-2l+4}\alpha_0\alpha_2) \\ & < \alpha_0(1 - 2^{-l+1} + 2^{-2l}) + \alpha_1(2^{-l+2}(2\alpha_0) + 2^{-2l+4}\alpha_0\alpha_1) \\ & < \alpha_0(1 - 2^{-l+1} + 2^{-2l} + \alpha_1(2^{-l+3} + 2^{-2l+3} + 2^{-2l+4}\alpha_1)) = B\ . \end{aligned}$$

□

Lemma 8 *If for some $n \in \mathbb{N}$, $\eta_n = 2^{-n}\frac{p_{n,n}}{p_{n,n+1}}$ and $\eta_{n+1} = 2^{-n-1}\frac{p_{n+1,n+1}}{p_{n+1,n+2}}$, then $\eta_n < \eta_{n+1}$.*

Proof:

$$\begin{aligned} 0 = p_{n+1,n+1} - 2^{n+1}\eta_{n+1}p_{n+1,n+2} = p_{n,n+1}(1 - 2^{-n-1}) + p_{n,n}2^{-n} \\ - \eta_{n+1}p_{n,n+1} = p_{n,n+1}(1 - 2^{-n-1} + \eta_n - \eta_{n+1})\ , \end{aligned}$$

but $1 - 2^{-n-1} > 0$, so $\eta_n < \eta_{n+1}$. □